**WISSEN & HINTERGRÜNDE • VIELE INFOGRAFIKEN • ILLUSTRATIONEN • FÜR EINSTEIGER & PROFIS**

PC Games Hardware Wissen

# So funktioniert Ihr Computer

2019/2020
*Aktualisierte vierte Auflage*

€ 11,99 08/2019
Österreich 13,50 € | Schweiz sfr 20,00 | Holland/Belgien/Luxemburg € 13,90
4 198581 811996 08
COMPUTEC MARQUARD MEDIA GROUP
EDITION
Hardware

# DAS TEST-MAGAZIN
# FÜR DIGITALE UNTERHALTUNG

**DAS BESTE AUS ALLEN TECHNIK-WELTEN:**

Flat-TVs | Smartphones | Tablets | Notebooks | Digitalkameras | Video | HiFi

plus: **DIE COOLSTEN SPIELE** und **ALLE FILM-BLOCKBUSTER**

**2 TOP-MOVIES AUF HEFT-DVD**

**Auch als Magazin-Variante ohne DVD für € 4,90**

**WWW.SPIELEFILMETECHNIK.DE**

**SFT bequem online bestellen:**
**shop.spielefilmetechnik.de**

**Oder einfach digital lesen:**
**epaper.spielefilmetechnik.de**

PC Games Hardware Wissen

# So funktioniert Ihr Computer

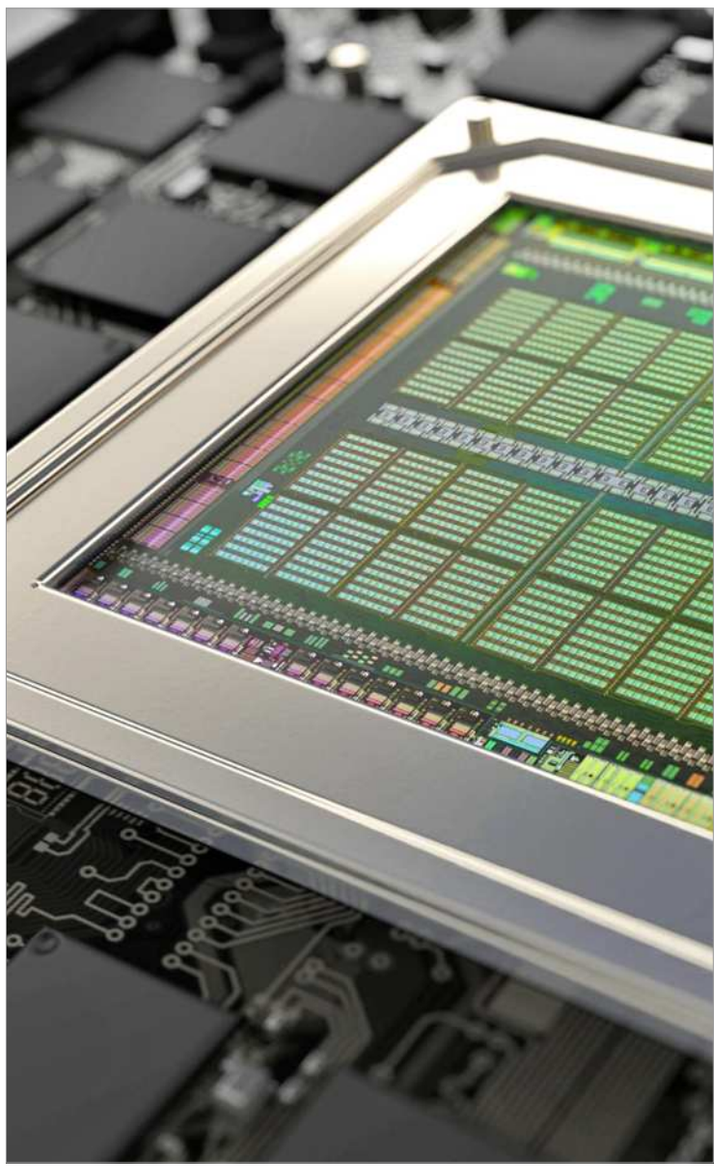
Willkommen zur vierten Ausgabe unseres Wissens-Kompendiums „So funktioniert Ihr Computer“!

Seit Erscheinen des vergangenen „Bookazines“ 2017 hat sich wieder eine Menge getan: AMD konnte mit Ryzen ein grandioses Comeback feiern, Deep-Learning-Algorithmen machen nun auch nicht mehr vor Spielen halt und die einst größte Computermesse der Welt, die CeBIT in Hannover, wurde sang- und klanglos eingestellt.

Um den Entwicklungen Rechnung zu tragen, haben wir in der 2019er-Edition unser Augenmerk verstärkt auf die Bereiche Grafikkarten und Prozessoren gelenkt; beide sind deutlich gewachsen und in weiten Teilen erstmals in dieser Reihe zu lesen. Neu ist ebenfalls der Software-Teil, in dem wir maschinelles Lernen und Kryptowährungen erklären – beides die Trendthemen der vergangenen beiden Jahre schlechthin.

Ich wünsche Ihnen viel Spaß beim Lesen und hoffe, Sie auch beim nächsten großen Update in zwei Jahren wiederzusehen, herzlichst Ihr

Christian Gögelein



# Inhalt

## Prozessoren


Startseite ..... 7
Überblick ..... 8
Schematische Darstellung ..... 12
Grundprinzip eines Prozessors ..... 14
So rechnen Prozessoren binär ..... 16
So rechnen Prozessoren komplexer .. 21
Register und Flip-Flops ..... 24
Pipelining, Out-of-Order-Execution .... 27
Caches ..... 30
Befehlssätze und Erweiterungen ..... 33
SMP und SMT ..... 37
Multi- und Hyper-Threading ..... 42
Mehrkern-Prozessoren ..... 44


## Grafikkarten


Startseite ..... 47
Überblick ..... 48
Schematische Darstellung ..... 50
Rastergrafik und Raytracing ..... 60
Grafikeffekte: Reflexionen ..... 66
Grafikeffekte: Umgebungsverdeck..... 70
Grafikeffekte: Texturdarstellung..... 73
Grafikeffekte: Kantenglättung..... 76
Grafikeffekte: Tiefenschärfe ..... 83
Grafikeffekte: Globale Beleuchtung... 86
Grafikeffekte: Voxel ..... 89
Grafikeffekte: Schatten..... 92


## Mainboard

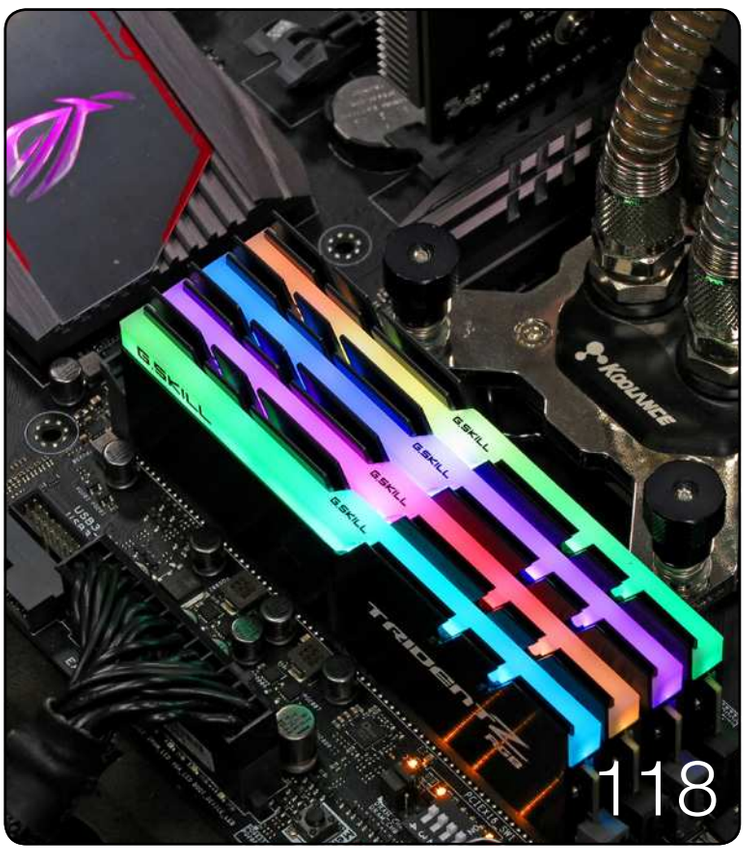

Startseite ..... 97
Überblick ..... 98
Schematische Darstellung ..... 100
Anschlüsse ..... 102
PCI-Express ..... 104
USB ..... 110
SATA ..... 115
Speicher ..... 118


## Infrastruktur


Startseite ..... 129
Solid State Disks (SSD) ..... 130
Festplatten (HDD) ..... 133
Netzteil ..... 138
Soundkarten ..... 146
Maus ..... 148
Tastatur ..... 150
VR-Brillen ..... 153


## Software


Startseite ..... 155
Künstliche Intelligenz und maschinelles Lernen ..... 156
Kryptowährungen ..... 160




COMPUTEC
MARQUARD MEDIA GROUP

Ein Unternehmen der Marquard Media Group AG
Verleger Jürg Marquard

**Verlag** Computec Media GmbH
Dr.-Mack-Straße 83, 90762 Fürth
Telefon: 0911/2872-100
leserpost@pcgameshardware.de
www.pcgameshardware.de | www.pcghx.de

**Geschäftsführer** Christian Müller, Rainer Rosenbusch

**Brand / Editorial Director (V.i.S.d.P.)** Thilo Bayer (tb), verantwortlich für den redaktionellen Inhalt, Adresse siehe Verlagsanschrift

**Leitende Redakteure** Raffael Vötter (Print, rv), Andreas Link (Online, al)
**Redaktion** Alexandros Bikoulis (ab), Manuel Christa (mc), Daniel Hoffmann (dh), Philipp Reuther (pr), Frank Stöwer (fs), Willi Tiefel (wt), Torsten Vogel (tv), Stephan Wilke (sw), Mark Mantel (mm)
**Mitarbeiter dieser Ausgabe** Roland Austinat (ra), Roman Hartung (rh), Christian Gögelein (cg)
**Leitung dieser Ausgabe** Christian Gögelein (cg)
**Lektorat** Claudia Brose (Ltg.), Birgit Bauer, Karline Folkendt
**Layout** Frank Pfründer (Ltg.), Hans Strobel
**Layoutkoordination** Albert Kraus
**Titelgestaltung** Frank Pfründer
**Fotografie Heft** tb, ab, al, mc, mm, pr, fs, wt, rv, tv, dw, sw, fp, hs
**Bildnachweis** Sofern nicht anders angegeben: PC Games Hardware
**DVD-Produktion** Thomas Dziewiszek

**Vertrieb, Abonnement** Werner Spachmüller (Ltg.)
**Marketing** Jeanette Haag (Ltg.), Simon Schmid
**Produktion** Martin Closmann

**www.pcgameshardware.de**

**Brand / Editorial Director** Thilo Bayer
**Redaktion** Andreas Link (Ltg.), Mark Mantel, Manuel Christa, Alexandros Bikoulis, Philipp Reuther, Frank Stöwer, Willi Tiefel, Torsten Vogel, Raffael Vötter, Stephan Wilke
**Digital Director** Simon Fistrich
**Entwicklung** Markus Wollny (Ltg.), René Giering, Aykut Arik, Tobias Hartlehnert, Christian Zamora, David Turkadze, Ali Adlah
**Webdesign** Tony von Biedenfeld, Daniel Popa
**SEO/Produktmanagement** Stefan Wölfel, Jennifer Brechtelsbauer

**Anzeigen**
CMS Media Services GmbH, Dr.-Mack-Straße 83, 90762 Fürth

**Sales Director**
Jens-Ole Quiel: Tel.: +49 911 2872-253; jens-ole.quiel@computec.de

**Anzeigenberatung Print / Digital**
Judith Gratias-Klamt: Tel.: +49 911 2872-252; judith.gratias-klamt@computec.de
Alto Mair: Tel.: +49 911 2872-144; alto.mair@computec.de
Bernhard Nusser: Tel.: +49 911 2872-254; bernhard.nusser@computec.de

**Verantwortlich für den Anzeigenteil**
Annett Heinze, Adresse siehe Verlagsanschrift

**E-Commerce & Affiliate** Daniel Waadt (Head of E-Commerce & Affiliate), Veronika Maucher
**Creation & Services** Wolfgang Fischer (Head of Creation & Services), Jan Weingarten
**Corporate Sales & Publishing** Dirk Gooding (Head of Corporate Sales & Publishing), Uwe Hönig

**Anzeigenberatung Online** Ströer Digital Media GmbH, Kehrwieder 8-9, 20457 Hamburg
Tel.: +49 40 – 468 567-100
www.stroeer.de, kontakt@stroeer.de

**Anzeigendisposition** Franziska Behme, Annett Heinze, anzeigen@computec.de
**Datenübertragung** via E-Mail: anzeigen@computec.de

Es gelten die Mediadaten Nr. 32 (01/2019)

**Abonnement – http://abo.pcgameshardware.de**
**Die Abwicklung (Rechnungsstellung, Zahlungsabwicklung und Versand) erfolgt durch DPV Deutscher Pressevertrieb GmbH als leistender Unternehmer.**

**Post-Adresse:**
Leserservice Computec, 20080 Hamburg, Deutschland
**Ansprechpartner für Reklamationen ist Ihr Computec-Team unter:**

**Deutschland**
E-Mail: computec@dpv.de, Tel.: 0911-99399098, Fax: 01805-8618002*
**Support:** Montag 07:00–20:00 Uhr, Dienstag–Freitag 07:30–20:00 Uhr, Samstag 09:00–14:00 Uhr
* (0,14 €/Min. aus dem dt. Festnetz, max. 0,42 €/Min. aus dem dt. Mobilfunk)

**Österreich, Schweiz und weitere Länder:**
E-Mail: computec@dpv.de, Tel: +49-911-99399098, Fax: +49-1805-8618002
**Support:** Montag 07:00–20:00 Uhr, Dienstag–Freitag 07:30–20:00 Uhr, Samstag 09:00–14:00 Uhr

Abonnementpreis für 12 Ausgaben: Magazin € 48,00 (€ 56,00 Österreich, € 60,00 Ausland), DVD € 62,- (€ 70,00 Österreich, € 74,- Ausland)

**Einzelversand/Nachbestellung**
Online: www.pcgh.de/go/shop, E-Mail: computec@dpv.de, Tel.: 0911-99399098 , Fax: 01805-8618002*
* (0,14 €/Min. aus dem dt. Festnetz, max. 0,42 €/Min. aus dem dt. Mobilfunk)

**ISSN/Vertriebskennzeichen PC Games Hardware:** 1616-6922 / B 53384

**Vertrieb:** DPV Deutscher Pressevertrieb GmbH, Am Sandtorkai 74, 20457 Hamburg, Internet: www.dpv.de
**Druck:** Walstead Central Europe, Poland, ul. Obr. Modlina 11, 30-733 Krakau, Polen

COMPUTEC MEDIA ist nicht verantwortlich für die inhaltliche Richtigkeit der Anzeigen und übernimmt keinerlei Verantwortung für in Anzeigen dargestellte Produkte und Dienstleistungen. Die Veröffentlichung von Anzeigen setzt nicht die Billigung der angebotenen Produkte und Service-Leistungen durch COMPUTEC MEDIA voraus. Sollten Sie Beschwerden zu einem unserer Anzeigenkunden, seinen Produkten oder Dienstleistungen haben, möchten wir Sie bitten, uns dies schriftlich mitzuteilen. Schreiben Sie unter Angabe des Magazins, in dem die Anzeige erschienen ist, inkl. der Ausgabe und der Seitennummer an: CMS Media Services GmbH, Franziska Behme, Anschrift siehe oben.

**Einsendungen Manuskripte und Programme:**
Mit der Einsendung von Manuskripten jeder Art gibt der Verfasser die Zustimmung zur Veröffentlichung in den von der Verlagsgruppe herausgegebenen Publikationen. Urheberrecht: Alle in PCGH veröffentlichten Beiträge bzw. Datenträger sind urheberrechtlich geschützt. Jegliche Reproduktion oder Nutzung bedarf der vorherigen, ausdrücklichen und schriftlichen Genehmigung des Verlags.

MARQUARD MEDIA GROUP

**Deutschland:**
4PLAYERS, AREAMOBILE, BUFFED, GAMESWORLD, GAMEZONE, GOLEM, LINUX-COMMUNITY, LINUX-MAGAZIN, LINUXUSER, MAKING GAMES, N-ZONE, GAMES AKTUELL, PC GAMES, PC GAMES HARDWARE, PC GAMES MMORE, PLAY 4, RASPBERRY PI GEEK, SFT, VIDEOGAMESZONE, WIDESCREEN

**Marquard Media Polska:**
CKM, COSMOPOLITAN, ESQUIRE, HARPER'S BAZAAR, JOY, KOZACZEK, PAPILOT, PLAYBOY, ZEBERKA

**Marquard Media Hungary:**
APA, ÉVA, GYEREKLÉLEK, INSTYLE, JOY, MEN'S HEALTH, PLAYBOY, RUNNER'S WORLD, SHAPE

# PCGH Digital im Abo

# Jeden Monat neue Prämienvorteile!

**Lesen Sie PCGH bequem am PC als Download-PDF, im Web oder auf Ihrem iOS-, Android- oder Amazon-Gerät!**

## 1-Jahres-Abo

**12 x PCGH**

amazon.de

**+ 10-Euro-Gutschein von Amazon.de**

www.pcgh.de/digital-only *

***1-Jahres-Abo: 39,99 Euro**

## 2-Jahres-Abo (NEU!)

**24 x PCGH**

**+ 1 brandaktuelles PC-Spiel als Key von Gamesplanet**

www.pcgh.de/digital2 **

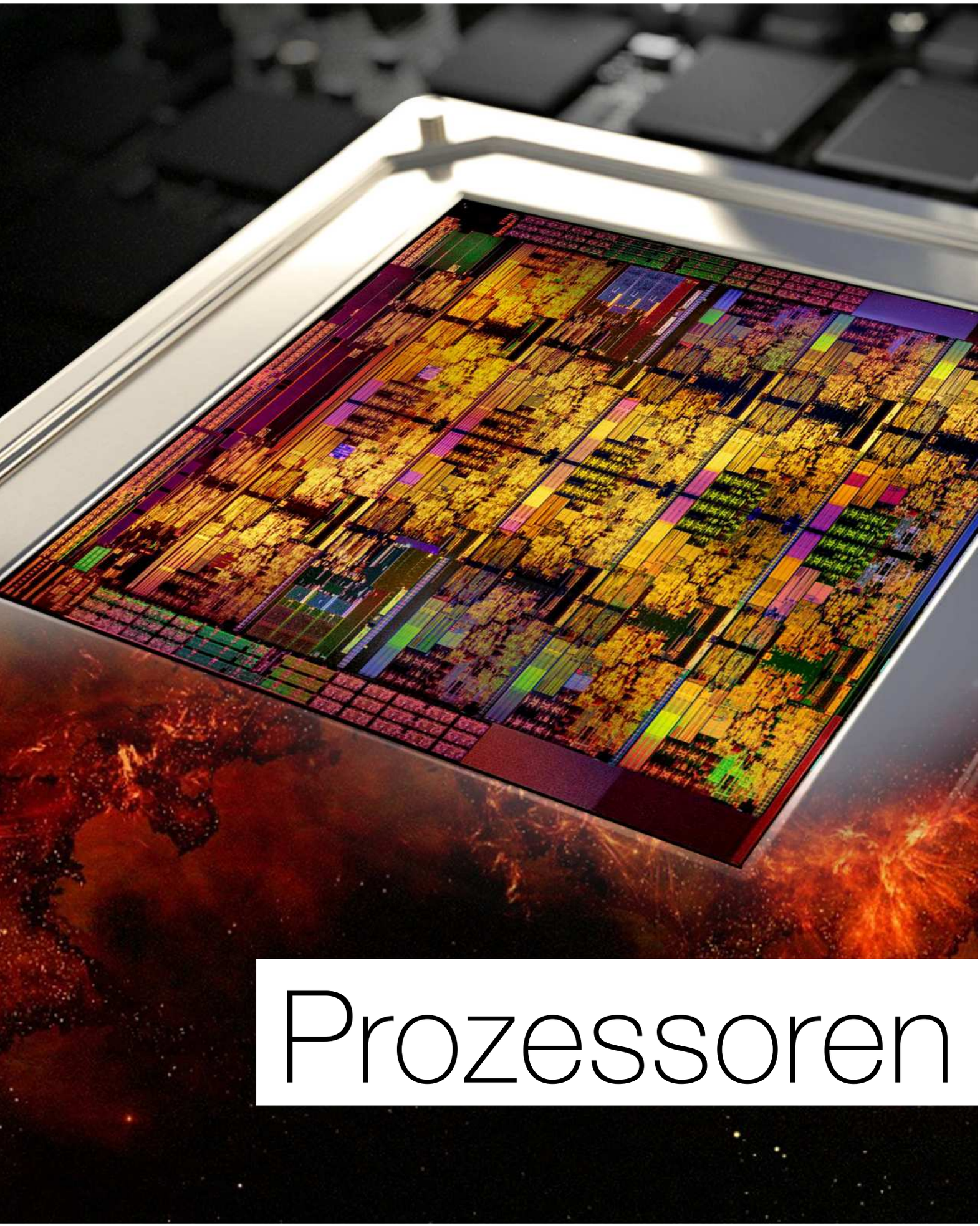
****2-Jahres-Abo: 79,98 Euro (In Deutschland)**

## Weitere Vorteile

- Werbefreier Zugang auf PCGH.de
- PCGH-Jahresarchive ab 2001 (PDF)
- Printartikel direkt kostenfrei im Webseitenlayout lesen – auf Desktop und Mobile!

PCGH +

**www.pcgh.de/digitalabo**



# Prozessoren

# So funktioniert ein Prozessor

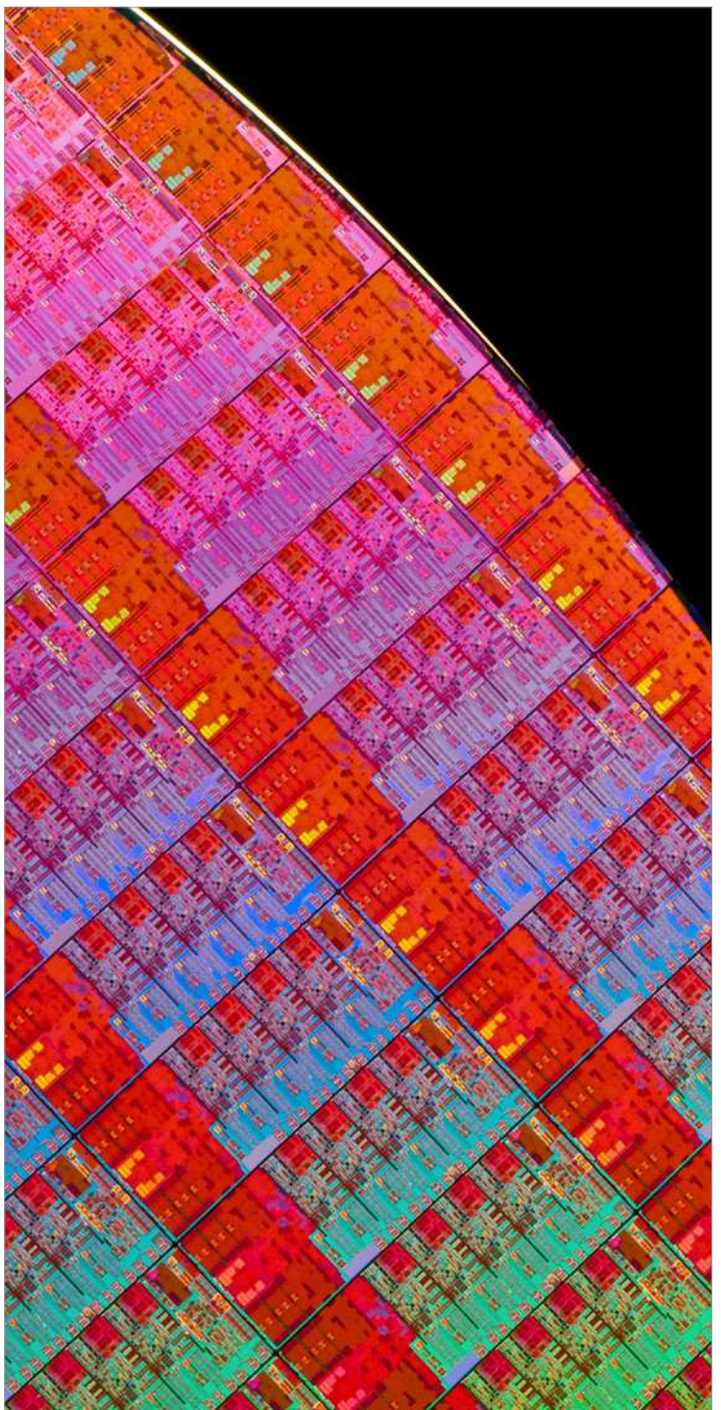


Bild: Intel

Prozessoren mit Dutzenden Kernen lassen sich zwar relativ leicht produzieren (wie hier im Bild bei einem Prototyp), die Kerne selbst enthalten aber nur einen Bruchteil der Funktionen vergleichbarer aktueller CPUs.

Prozessoren werden Jahr für Jahr immer komplexer und leistungsfähiger – aus den einfachen Schaltkreisen, die zu Beginn der PC-Ära Standard waren, sind inzwischen High-tech-Chips geworden, die hunderte Millionen Transistoren enthalten. Wie aus den Schaltern eine komplexe Rechenmaschine wird, zeigt dieses Kapitel; zunächst jedoch ein kurzer Überblick der Funktionsgruppen.

## Prozessoraufbau

Ein aktueller Prozessor besteht aus mehreren Baugruppen, die unterschiedlich viel Platz einnehmen. Die eigentlichen Rechenwerke belegen oft nur einen Bruchteil der Chipfläche. Der restliche Platz wird für verschiedene Funktionen benötigt, ein großer Teil des Transistorbudgets geht dabei für Cache-Speicher drauf. Grundsätzlich verfügen Prozessoren über:

- Mindestens einen CPU-Kern mit den Recheneinheiten
- Zwischenspeicher (L1, L2, L3, L4)
- Ein- und Ausgabeeinheiten
- mindestens einen Speichercontroller (inkl. diverser Northbridge-Funktionen)
- Gegebenenfalls einen Grafikkern

## CPU intern

Die Recheneinheiten eines Prozessors sind relativ einfach gestrickt und funktionieren grundsätzlich immer noch so wie zu den Anfangszeiten der Computergeschichte. Durch bestimmte Verschaltungsmuster können elementare Operationen wie eine Addition durchgeführt werden. Durch Kopplung mehrerer dieser Einheiten lassen sich auch Multiplikationen, Divisionen und viele weitere arithmetische Operationen umsetzen.

## Befehlsverarbeitung

Das Prinzip der Befehlsverarbeitung hat sich ebenfalls nicht geändert: Ein Befehl wird zunächst vom Prozessor geholt, dann dekodiert und ausgeführt. Schließlich wird das Ergebnis zurückgeschrieben. Gleichwohl hat sich die Geschwindigkeit der Abarbeitung in den vergangenen Jahrzehnten natürlich drastisch erhöht.

Doch was ist eigentlich ein „Befehl“? Mit dem ersten x86-Prozessor führte Intel den gleichnamigen x86-Befehlssatz ein. Darin waren (und sind bis heute) 80 Instruktionen enthalten, die hauptsächlich für die Berechnung von mathematischen Aufgaben, aber auch für die Manipulation von Zeichenketten oder Sprüngen im Code verwendet werden. Einfache Instruktionen heißen zum Beispiel ADD (Addition), MOV (Move, zum Bewegen von Daten in Speicher bzw. Register) oder CMP (Compare, vergleicht Operanden).

## Arbeitsschritte

Befehle lagen früher in Form von Lochkarten vor, sind heute aber in der Regel im Arbeitsspeicher oder auf Festspeichern wie einer Festplatte vorhanden. Weil aber selbst der Arbeitsspeicher noch bis zu 1.000-mal langsamer ist, als die CPU intern rechnen kann, setzen die Entwickler auf chipinterne Zwischenspeicher, die häufig benötigte Daten „cachen“. Damit wird das zeitraubende Laden von Daten aus dem Arbeitsspeicher häufig vermieden.

Die Befehlsdekodierung läuft je nach Prozessor unterschiedlich ab. Grundsätzlich wird dabei ermittelt, welche Operation durchgeführt

werden soll (Addition, Division, Sprung etc.). Zusätzlich werden die Speicheradressen festgelegt, in denen sich die zu verarbeitenden Daten befinden. Intel und AMD zerlegen die standardisierten x86-Operationen dabei zusätzlich in kleinere, maschinengerechtere Befehlsgruppen.

## Micro- und Makro-Ops

Intel nennt diese Gruppen Mikro-Instruktionen („Micro-Ops"), bei AMD hießen sie lange Zeit Makro-Instruktionen („Makro-Ops"); erst in der jüngeren Vergangenheit schwenkte auch AMD auf die Bezeichnung Micro-Ops um.

Seit Einführung dieser schlanken Instruktionen (mit dem Pentium Pro) gelten nun auch x86-CPUs als RISC-Prozessoren. RISC-CPUs verfügen über einen reduzierten Befehlssatz („Reduced Instruction Set Computer"), sind dabei aber deutlich schneller als solche mit einem komplexen Befehlssatz („Complex Instruction Set Computer", „CISC")

Sind alle Daten vorhanden, wird der Befehl ausgeführt. Die Operanden werden von einer Ausführungseinheit (Arithmetic Logic Unit, kurz: ALU) verarbeitet; aus zwei Operanden wird so zum Beispiel die Summe gebildet. Schließlich wird das Ergebnis in den (Cache-)Speicher zurückgeschrieben. Der Befehlszähler („Program Counter") wird erhöht und die Befehlsverarbeitung wird mit der nächsten Operation fortgesetzt – so lange, bis das Programm beendet ist.

## Rechenwerke

Die Rechenwerke eines Prozessors waren primär für Operationen mit Ganzzahlen ausgelegt. Viele Anwendungen, darunter besonders Spiele, haben aber einen hohen Anteil an Berechnungen mit Kommazahlen (Gleitkommaoperationen). Schon früh konstruierten die Chip-Ingenieure daher spezielle Gleitkomma-Einheiten. Anfangs steckten diese noch in eigenen Gehäusen und mussten über das Mainboard mit dem Prozessor verbunden werden. Seit der 486-Generation sind diese Teil des Hauptprozessors.

Bald stieß auch die Gleitkommaeinheit an ihre Grenzen. Intel führte mit dem Pentium III deshalb erstmals eine SSE-Einheit ein. Diese Einheit ist deshalb besonders interessant, weil sie auf einen eigenen Registersatz zurückgreift, der viermal breiter ist als bei einem 32-Bit-Prozessor. So wird es möglich, mit einer Instruktion mehrere Daten zu verarbeiten, was bei einigen Aufgaben deutliche Leistungsgewinne bringt. Soll zum Beispiel die Helligkeit eines Bildes erhöht werden, kann eine Instruktion (eine Addition auf den Farbwert der RGB-Kanäle) auf verschiedene Daten (die Pixel des Bildes) angewendet werden – mit nahezu vierfacher Geschwindigkeit.

## Cache-Hierarchie

Während seiner Arbeit muss der Prozessor ständig auf den Arbeitsspeicher (RAM) zugreifen. Dieser ist zwar sehr preiswert, aber im Vergleich zum Prozessor sehr langsam. Eine ausgefeilte Cache-Hierarchie sorgt daher bei Prozessoren für eine deutlich reduzierte Anzahl von Speicheranfragen. Der schnellste und leistungsfähige Cache stellt Daten in der Regel innerhalb von drei bis vier Taktzyklen bereit. Er heißt Level-1(L1)-Cache und wird nur sparsam eingesetzt. Aktuelle Prozessoren verfügen pro Kern über 64 bis 128 Kilobyte L1-Cache, der in einen Daten- und einen Befehlscache unterteilt ist.

An zweiter Stelle steht der L2-Cache, der üblicherweise 256 bis 512 Kilobyte groß ist. Er ist mit zirca 10 bis

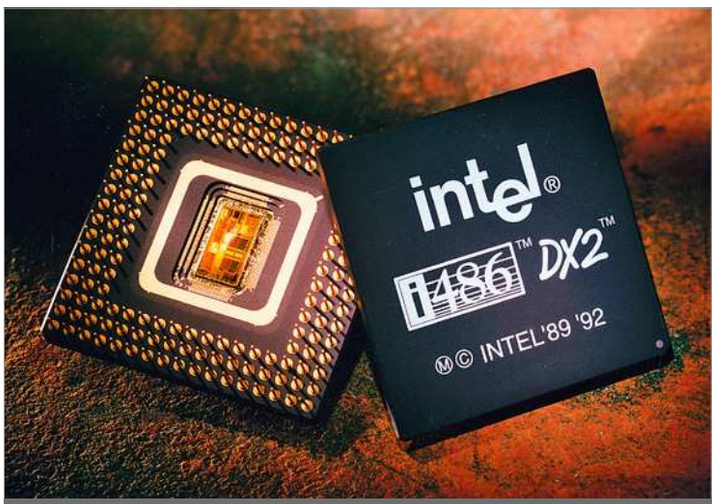



Bild: Intel

Mit Intels 486 hielten Gleitkommaeinheiten Einzug in die Prozessor-Architektur. Dies verhalf auch (3D-)Spielen zum Durchbruch.

Bild: Intel

Intels Pentium Pro brachte eine Reihe neuer Funktionen, unter anderem die Befehlszerlegung in Micro-Ops.

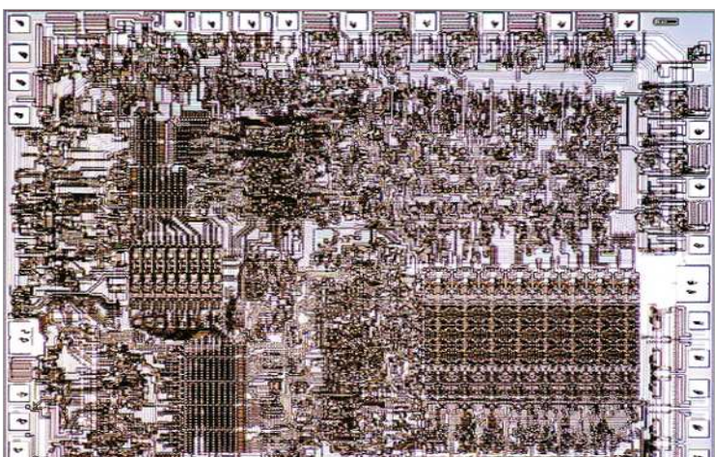


Bild: Intel

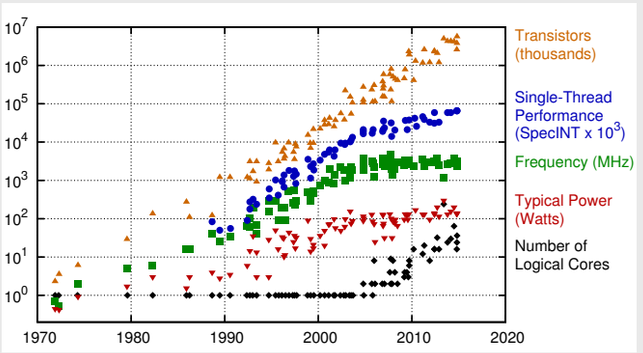
Noch relativ leicht zu überblicken war ein 8088-Prozessor mit knapp 29.000 Transistoren. Mit jeder neuen Generation kamen danach neue Funktionen und Techniken hinzu.

# Prozessor-Trends: Was bringt die Zukunft?

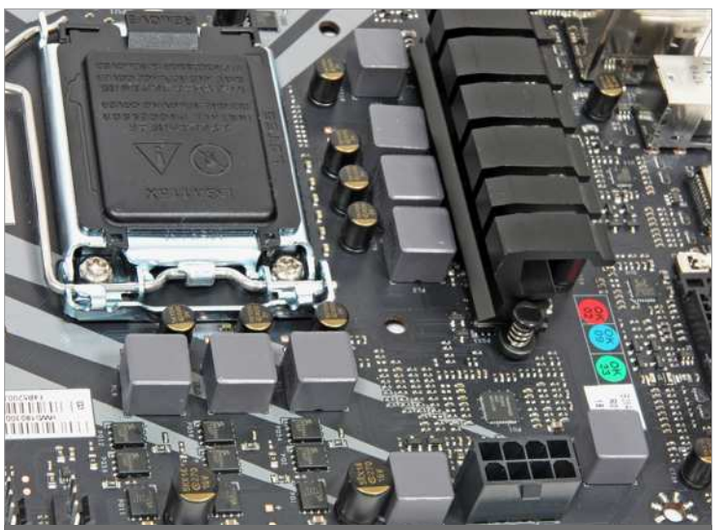
**Die Prozessorentwicklung wird in mehrfacher Hinsicht gebremst. Einerseits stoßen CPUs an ein Taktlimit, das bei annähernd 4 Gigahertz liegt. Knapp zehn Jahre lang konnte kein Prozessor die Schwelle von 3,8 GHz „knacken“ und auch danach ging es nur zäh voran. Andererseits kann auch die Zahl der CPU-Kerne nicht beliebig gesteigert werden – jedenfalls nicht, wenn „normale“ CPU-Kerne verwendet werden.**

**AMD und Intel wollen künftig auf abgespeckte Kerne setzen, welche die vollwertigen Kerne bei speziellen Rechenaufgaben unterstützen. Intel spricht konkret bereits von Kryptographie-Kernen. Integrierte Grafikkerne sind bereits Realität.**

**Wie sich die maßgeblichen CPU-Parameter Transistorzahl, Taktfrequenz und Stromverbrauch in den vergangenen Jahren entwickelt haben, illustriert die folgende Grafik:**




Grafik: Christopher Batten, mit Daten von M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond; lizenziert unter CC-BY-4.0 (https://creativecommons.org/licenses/by/4.0/)



Die niedrigen Spannungen aktueller Prozessoren benötigen aufwendige Spannungswandler auf Mainboards. Aktuelle CPUs verwenden intern sogar mehrere Spannungen.

20 Taktzyklen langsamer als der L1-Cache. Schließlich steht ein gemeinsamer Level-3-Cache für alle Kerne zur Verfügung. Er ist meist sechs bis acht Megabyte groß und mit 40 bis 100 Taktzyklen deutlich langsamer. Im Gegensatz zum DRAM-Arbeitsspeicher ist Cache-Speicher aus SRAM-Zellen zusammengesetzt. Diese bestehen aus vier bis sechs Transistoren und benötigen mehr Die-Fläche als DRAM; dafür können Informationen dauerhaft gespeichert werden und müssen nicht in festen Intervallen sogenannte Refresh-Zyklen durchlaufen.

## Pipelining und OoO

Die immer größer werdenden Caches haben die Verarbeitung von Daten zwar stark beschleunigt. Sie können aber nicht verhindern, dass manche Daten immer noch aus dem Speicher oder – noch schlimmer – von der Festplatte geholt werden müssen. Um die Recheneinheiten besser auszulasten, verfügen alle modernen x86-Prozessoren seit dem Pentium über „Pipelines“; gemeint ist damit, dass mehrere Befehle parallel abgearbeitet werden. Während ein Befehl ausgeführt wird, wird ein weiterer schon dekodiert und ein dritter geholt.

Pipelining bringt vor allem dann Vorteile, wenn Befehle unabhängig voneinander ausgeführt werden können; dies ist jedoch nur selten der Fall. Viel häufiger kommt es vor, dass ein Befehl auf das Ergebnis einer anderen Operation warten muss. Um die „Wartezeit“ sinnvoll zu nutzen, haben AMD und Intel eine sogenannte „Out-of-Order“(OoO)-Execution ersonnen. Damit lassen sich Befehle auch spekulativ ausführen, was im Erfolgsfall einige Taktzyklen spart.

## Kommunikationswege

Ein Prozessor alleine könnte wenig ausrichten. Mindestens ebenso wichtig ist die Anbindung an den Rest des Systems. Während der Anwender beispielsweise ein Spiel spielt, werden in einer zentralen Schleife permanent wichtige Parameter wie Maus- und Tastatureingaben abgefragt, Daten aus dem Speicher geholt, verarbeitet, an die Grafikkarte gesendet oder zurückgeschrieben und viele weitere Aufgaben erledigt. Bis vor einigen Jahren wurde für den kompletten Datenverkehr von und zum Prozessor ein eigener Bus verwendet, der „Front Side Bus“. Diese Verbindung wurde jedoch immer mehr zum Flaschenhals und daher erst von AMD, später auch von Intel durch eine schnellere Punkt-zu-Punkt-Verbindung ersetzt. Die Speichermodule werden zusätzlich von einem speziellen On-Die-Controller angesprochen, was die Wartezeit weiter verkürzt hat.

## Leistungsaufnahme

Früher liefen Prozessoren noch mit der einheitlichen I/O-Spannung des Mainboards (5 bzw. 3,3 Volt). Aufgrund der stetig steigenden Verlustleistung sind die Hersteller jedoch bemüht, die CPU-Spannung weiter zu senken. Als Folge dieser Anstrengungen laufen heutige Prozessoren mit Spannungen um ein Volt, die vom Mainboard eigens bereitgestellt werden müssen. Das hat zur Folge, dass die Mainboard-Hersteller Spannungswandler integrieren müssen, die zusätzlich Wärme produzieren.

Die Leistungsaufnahme der Prozessoren steigt ebenfalls mit jedem neuen Transistor. Die Hersteller versuchen, dies (neben feineren Herstellungsprozessen) durch Energiesparfunktionen wie Cool’n’quiet oder EIST aufzufangen. Dennoch bleibt das grundsätzliche Problem bestehen: Bei jedem Transistor-Schaltvorgang wird Energie „verbraucht“. Würden alle Transistoren eines Prozessors zur gleichen Zeit schalten, würde der Chip augenblicklich durchbrennen!

★ KOMPETENT ★ VIELSEITIG ★ KRITISCH ★

# DEUTSCHLANDS ERSTAUNLICH EHRLICHES FILMMAGAZIN

WIDESCREEN – DAS BLU-RAY-, DVD- UND KINO-MAGAZIN

AUCH DIGITAL ERHÄLTLICH | **WWW.WIDESCREEN-ONLINE.DE**

# Der Weg der Daten durch den Prozessor

## Vorbereitung 1

Programme werden normalerweise linear abgearbeitet. Ein spezielles Register, der Program Counter (PC), hält die Adresse bereit, an welcher jener Befehl steht, der als Nächstes bearbeitet werden soll. Der PC wird nach jeder Instruktion erhöht, bis das Programm beendet ist.

Am Anfang unseres Beispiels wird also der erste (x86-)Befehl geholt, er soll bereits im Speicher (RAM) liegen. Dazu wird die entsprechende Adresse an den Speicherkontroller gesendet, wo die entsprechenden Zeilen und Spalten adressiert werden müssen. Bis die Daten über den Bus geschickt werden und dort ankommen, vergehen dutzende bis mehrere hundert CPU-Taktzyklen. Muss das Programm von der Festplatte geholt werden, vervielfacht sich die Wartezeit.

Meist werden gleichzeitig die benachbarten Speicherzellen mit ausgelesen und in eine Cacheline des Prozessors gelegt. Bei der Bearbeitung eines Programms liegen die Befehle normalerweise hintereinander; die folgenden Befehle sind dann schon im (Instruktions-) Cache vorhanden und können wesentlich schneller bereitgestellt werden.

## Befehl analysieren 2

Der x86-Befehl wird nun dekodiert und in kleinere Instruktionen zerlegt. Diese kleineren Instruktionen haben den Vorteil, dass sie alle einfach gehalten sind, die gleiche Länge haben und damit schnell ausgeführt werden können. Bei einigen Prozessoren werden die bereits dekodierten Befehle in einem speziellen Cache gespeichert. Beim Pentium 4 hieß dieser „Trace Cache“, Coffee Lake verfügt über einen ähnlich gestrickten „Micro-Op-Cache“, der mehrere tausende dekodierte Mikro-Instruktionen aufnehmen kann. Bei Intel-Architekturen wird zwischen einfachen und komplexen x86-Befehlen unterschieden; während für einfache Instruktionen drei Dekoder bereitstehen, gibt es für komplexe Anweisungen nur einen. Bei der Befehlsdekodierung wird außerdem festgelegt, ob und welche Befehle sich umsortieren lassen. Durch die „Out-of-Order-Execution“ können die Einheiten der CPU besser ausgelastet werden.

## Ausführung vorbereiten 3

Der Befehl ist jetzt dekodiert und zerlegt – idealerweise können die Operanden gleich mit an die Ausführungseinheiten übergeben werden. Ein Befehl der Art „Addiere 20 zu 6“ könnte also sofort verarbeitet werden. In der Regel wird ein Compiler versuchen, die benötigten Daten im Voraus bereitzustellen. Wenn die Operanden aber von dem Ergebnis einer vorhergehenden Instruktion abhängig sind, klappt dies nicht. Bis die Daten eintreffen, verharrt der Befehl in diesem Fall in der „Reservation Station“. Je nachdem, welcher Befehl ausgeführt werden soll, nimmt die Operation nun unterschiedliche Wege (im Folgenden dargestellt als 4a, 4b und 4c).

## Befehl ausführen 4

Anschließend werden die Ausführungseinheiten für die Befehlsausführung vorbereitet. Konkret heißt das, dass die Recheneinheiten (ALUs) für die entsprechende Operation mit den zugehörigen Registern verschaltet werden. Je nach Architektur stehen unterschiedlich viele ALUs für diverse Aufgaben bereit. Üblich sind heute drei bis sechs ALUs für Ganzzahl- und/oder Speicherzugriffe sowie drei weitere ALUs für Fließpunktoperationen (diese heißen dann FPUs). Häufig setzen Chipentwickler auch hybride Rechenwerke ein; so gibt es dann beispielsweise mehrere Ausführungseinheiten für einfache Befehle, aber nur eine Einheit, die komplexe Befehle berechnen kann. Da komplexe Befehle selten gehäuft vorkommen, resultiert daraus unter dem Strich ein Performance-Plus.

4a

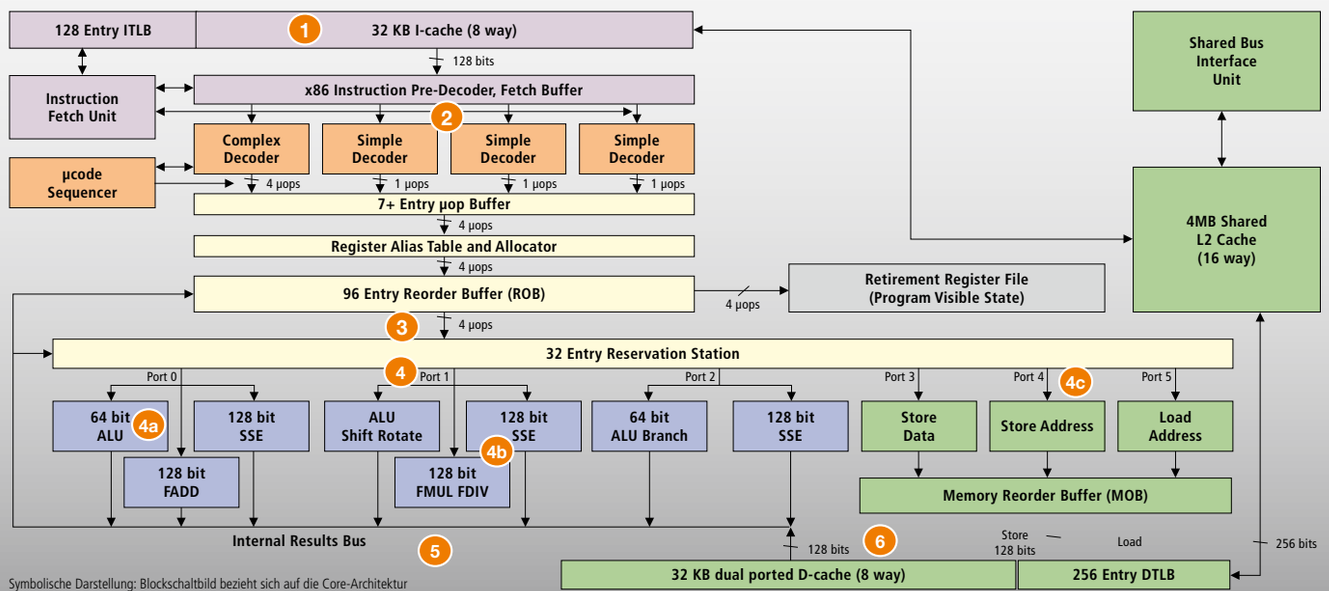
**Integer-Operation**
Im Fall 4a soll eine Ganzzahl verarbeitet werden; eine einfache Addition kann bei vielen Prozessoren von sechs verschiedenen Ausführungseinheiten berechnet werden.

4b

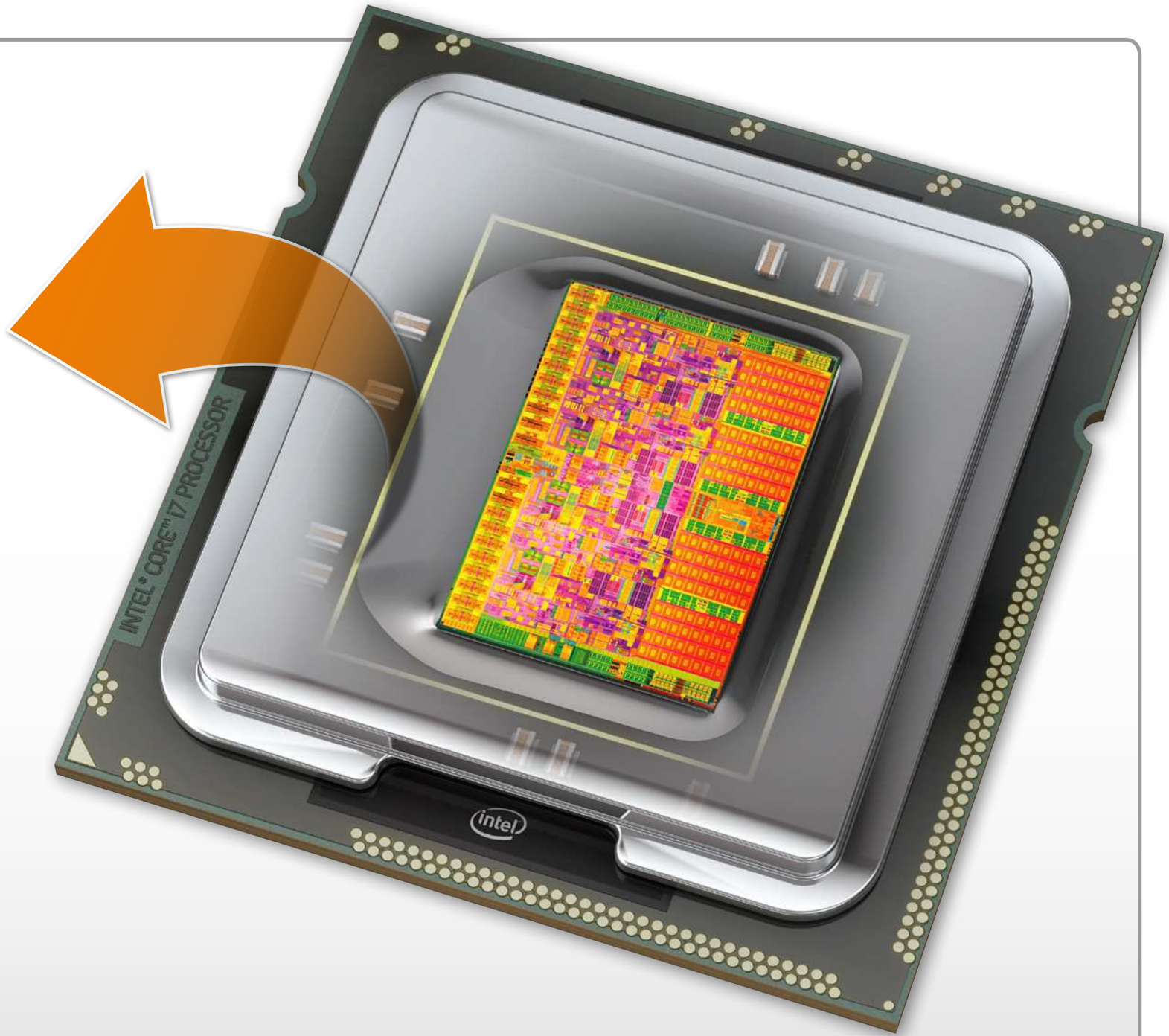
**Fließpunkt-/SSE-Operation**
Sollen bei dem Befehl Zahlen mit Nachkommastellen verarbeitet werden, wird dafür eine Fließpunkteinheit bemüht. In der Coffee-Lake-Architektur gibt es eigene Rechenwerke für Fließpunkt-Additionen, -Multiplikationen und -Divisionen. Für SSE-Befehle (bzw. MMX/3d-now) sind ebenfalls spezielle Ausführungseinheiten vorgesehen.

4c

**Speicher-Operation**
Oft müssen Daten vom und zum Speicher bewegt werden (siehe Schritt 3); dafür bieten moderne Prozessoren eigene Einheiten an. Zum Laden und Speichern gibt es meist drei Einheiten (AGUs).




Symbolische Darstellung: Blockschaltbild bezieht sich auf die Core-Architektur




## Ergebnis schreiben 5

Schließlich liegt das Resultat der Operation vor; es wird aber nicht direkt in den Cache geschrieben, sondern in die „Reservation Station“ geschickt. Dort steht das Ergebnis für nachfolgende Instruktionen zur Verfügung, was die Ausführungszeit verkürzt. Dies kommt ebenfalls bei OoO-Executions vor, um die „Wartezeiten“ sinnvoll zu nutzen.

Andere Befehle sorgen dafür, dass weitere Operanden geladen werden, und wieder andere können den Program Counter direkt verstellen – was dann als „Sprung“ bezeichnet wird. Die Ergebnisse der Speicher-Einheiten (AGU) gehen in einen Puffer, den Memory Reorder Buffer (MOB). Dort können die Zugriffe umsortiert werden. Das ist sinnvoll, damit schnelle Lesezugriffe den langsamen Schreibzugriffen vorgezogen werden können.

## Speicher aktualisieren 6

Hat die Rechenoperation ein Ergebnis zur Folge, so wird dieses schließlich in den Cache und/oder den Speicher geschrieben. Bei einigen BIOS-Versionen kann man das sogar steuern. „Write Back“ bedeutet dabei, dass der Wert nur in den Cache geschrieben wird. Mit „Write Through“ werden Cache und Speicher gleichermaßen aktualisiert, was etwas mehr Zeit kostet.

Generell sind Schreibzugriffe weniger performancekritisch als Lesezugriffe. Diverse Puffer und Algorithmen sorgen dafür, dass die aktualisierten Daten gebündelt und am Stück in den Speicher geschrieben werden.

# Das Grundprinzip eines Prozessors

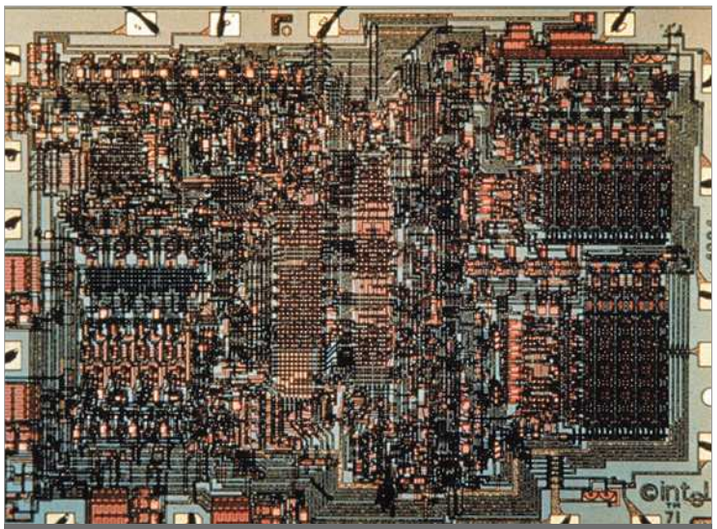
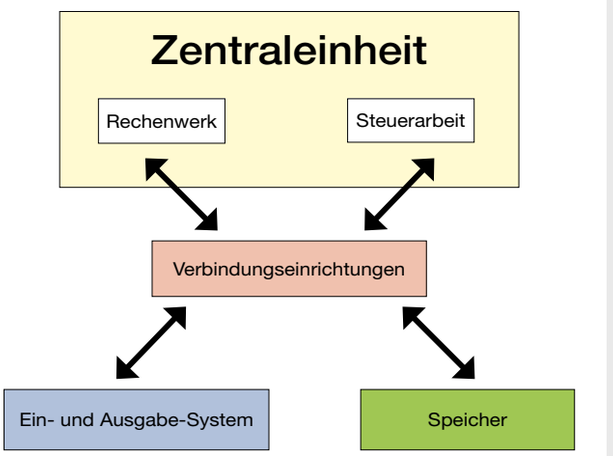


Bild: Intel

Prozessoren bestehen mittlerweile aus mehr als einer Milliarde Transistoren. Intels erster Mikroprozessor, der 4004 (hier im Bild), vereinte gerade mal 2.300 elektrische Schalter.

## Die Komponenten eines Von-Neumann-Rechners

**Das Grundprinzip eines Universalrechners nach von Neumann: Prinzipiell arbeiten auch aktuelle CPUs noch nach diesen Vorgaben.**




Auch modernste Prozessoren verarbeiten Befehle immer noch nach jenem Prinzip, welches der in Ungarn geborene Mathematiker John von Neumann bereits im Jahre 1945 konzipierte.

### Elementarer Aufbau

Sein früher theoretischer Rechner bestand aus den Funktionseinheiten Steuerwerk, Rechenwerk, Speicher, Eingabe- und Ausgabewerk. Per Definition ist die Struktur des Von-Neumann-Rechners unabhängig von den zu bearbeitenden Problemen, weshalb man eine solche Maschine auch Universalrechner nennt. Sie kann jedes beliebige Rechenproblem lösen und ist nicht an eine spezielle Aufgabe gebunden, wie es etwa ein Cola-Automat ist.

Zur Lösung eines Problems muss von außen eine Bearbeitungsvorschrift, das Programm, eingegeben und im Speicher abgelegt werden. Ohne dieses Programm ist die Maschine nicht arbeitsfähig. Programme, Daten, Zwischen- und Endergebnisse werden in demselben Speicher abgelegt. Der Speicher ist in gleich große Zellen unterteilt, die fortlaufend durchnummeriert sind. Über die Nummer (Adresse) einer Speicherzelle kann deren Inhalt abgerufen oder verändert werden. Aufeinanderfolgende Befehle eines Programms werden in aufeinanderfolgenden Speicherzellen abgelegt. Das Ansprechen des nächsten Befehls geschieht vom Steuerwerk aus durch Erhöhen der Befehlsadresse um Eins.

Durch Sprungbefehle kann von der Bearbeitung der Befehle in der gespeicherten Reihenfolge abgewichen werden. Es gibt arithmetische Befehle wie Addieren, Multiplizieren, Konstanten laden, logische Befehle wie Vergleiche, logisches NICHT, UND, ODER, Transportbefehle, zum Beispiel vom Speicher zum Rechenwerk und für die Ein-/Ausgabe, bedingte Sprünge und sonstige Befehle wie Schieben, Unterbrechen, Warten usw. Von einigen wenigen Ausnahmefällen abgesehen, orientieren sich die heutigen PCs immer noch an der Struktur des klassischen Universalrechners.

### Von-Neumann-Zyklus

Der Prozess der Befehlsverarbeitung bei Von-Neumann-Rechnern wird Von-Neumann-Zyklus genannt und besteht aus folgenden fünf nacheinander ablaufenden Teilschritten: FETCH, DECODE, FETCH OPERANDS, EXECUTE, UPDATE (INSTRUCTION POINTER). Grundsätzlich gehört das Zurückschreiben der Daten (WRITE oder WRITE BACK) auch noch dazu, der ursprüngliche Von-Neumann-Rechner kannte aber noch keine Zwischenspeicher und benötigte diese Schritte daher nicht.

Das grundlegende Prinzip dieser Befehlsverarbeitung ist bis heute unverändert geblieben, der Einfachheit halber wird es in CPU-Schaubildern manchmal auf die drei Phasen FETCH (engl.: „holen“), DECODE (engl.: „entschlüsseln“) und EXECUTE (eng.: „ausführen“) reduziert.

### Befehlsverarbeitung

Beim FETCH-Schritt wird aus RAM- oder ROM-Speicher der nächste zu bearbeitende Befehl geholt. Dieser Befehl wird bei DECODE durch das Steuerwerk in Schaltinstruktionen aufgelöst, die das Rechenwerk „verstehen“ kann. Aus RAM oder ROM werden nun bei FETCH OPERANDS

die Operanden geholt, also die Werte, die durch den Befehl verändert werden sollen bzw. die als Parameter verwendet werden, also etwa die beiden Operanden einer Addition. Bei EXECUTE wird die Operation vom Rechenwerk ausgeführt. Im Schritt UPDATE INSTRUCTION POINTER wird der Befehlszähler erhöht, damit der Rechner „weiß“, an welcher Stelle des Programms er sich gerade befindet. Das geschieht parallel zum DECODE und FETCH OPERANDS. Beim EXECUTE kann der Befehlszähler wieder verändert werden (Sprungbefehl). Anschließend beginnt der Zyklus in der Regel von vorn und der nächste Schritt des Programms wird ausgeführt.

## Schalter und Transistoren

Prozessoren arbeiten mit Strom, respektive mit elektrischen Schaltern, Transistoren genannt. Der Transistoreffekt wurde 1947 durch William Shockley, John Bardeen und Walter Brattain in den Bell Laboratories von der Firma AT&T entdeckt. Später erhielten sie dafür den Nobelpreis für Physik. Durch einen kleinen Steuerstrom kann in einem Transistor ein wesentlich größerer Strom gesteuert werden.

Der Transistor wurde auf der Grundlage der Diode entwickelt. Eine Diode besteht aus zwei dotierten Halbleiterschichten (NP- beziehungsweise PN-dotiert) und lässt Strom nur in einer Richtung durchfließen. Ein Transistor ist im Wesentlichen eine Zusammenschaltung aus drei Diodenhälften (NPN beziehungsweise PNP), wobei die obere und untere Schicht als Kollektor beziehungsweise Emitter und die mittlere Schicht als Basis bezeichnet werden.

## Bipolartransistor

Beim Bipolartransistor steuert ein Strom im Basis-Emitter-Kreis einen stärkeren Strom im Kollektor-Emitter-Kreis.

Der Vorteil dieser stark miniaturisierten Schalter: Es passen Tausende, ja Millionen davon auf einen einzigen Chip. Die Kombination aus diesen Millionen von Transistoren erlaubt die Darstellung von unzähligen Funktionen, die wiederum in ihrer Kombination den Befehlssatz eines Prozessors ausmachen.

## Das Binärsystem

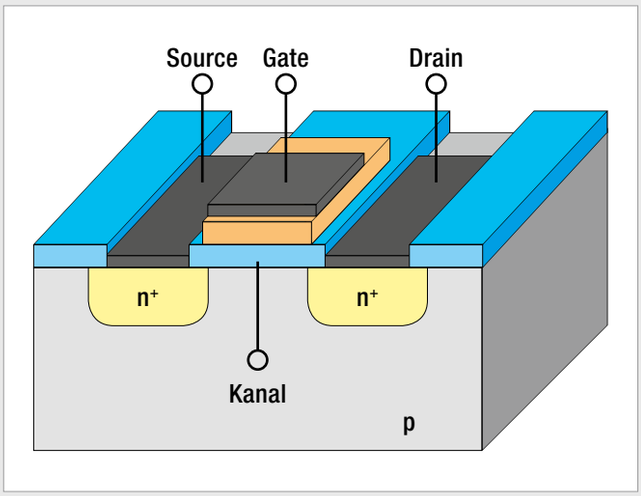
Das Binärsystem beschreibt ein Stellenwertsystem zur Darstellung von Zahlen auf der Basis 2. Warum ausgerechnet die 2? Wenn eine Informationseinheit wie beim Fluss von Strom nur zwei Zustände (Ein/Aus) annehmen kann, dann kann man Zahlen natürlich nicht im Dezimalsystem (Basis 10) darstellen, wie wir es normalerweise tun. Man muss sich auf das Binärsystem beschränken.

Während bei unserem Dezimalsystem zehn Zahlen (0 bis 9) möglich sind, ehe wir eine zusätzliche Stelle benötigen (10), sind es bei Dualzahlen nur zwei. Zählen würden wir demnach so: 0, 1, 10, 11, 100, 101, 110 usw., wobei die Binärzahl 10 im Dezimalsystem der 2 entspricht, die duale 11 der 3 usw. Es ändert sich also nichts an den Zahlen, sondern nur an der Schreibweise.

Mit diesen Dualzahlen im binären Code (daher auch Binärzahlen oder Zahlen im Binärcode genannt) kann eine aus Schaltungen bestehende Maschine nun arbeiten. Man muss nur festlegen, dass eine Eins dem Zustand „Strom fließt“ bzw. „Spannung angelegt“ entspricht, eine Null dem Zustand „Strom fließt nicht“ bzw. „Keine Spannung angelegt“. Wie genau das Rechnen mit Binärzahlen funktioniert, zeigen die folgenden Seiten.

# Wie ein Transistor Strom leitet

**Transistoren funktionieren wie Schalter: Liegt ein schwacher Strom an (Mitte, Gate), kann ein stärkerer Strom von der Quelle (Source, links) zur Ableitung fließen (Drain, rechts). Hier dargestellt ist ein in CPUs verwendeter Feldeffekttransistor.**




# Wie ein Transistor aufgebaut ist

**Der oben dargestellte Transistor ist ein Metalloxid-Halbleiter-Feldeffekttransistor. Dieser kann in zwei Arten realisiert werden: entweder der Kanal befördert positive („p-dotiert“) oder negative Ladungen („n-dotiert“). Bei p-dotierten Transistoren wurden entweder zusätzliche positive Ladungsträger in das Material (im Bild oben die graue Fläche unten) eingefügt oder dem Material wurden Elektronen entzogen, wodurch sogenannte Löcher entstehen.**

**Im „Normalzustand“ fließt in unserem Beispieltransistor kein Strom; denken Sie sich den blauen Kanal in der Mitte einfach weg oder – wenn es ganz genau sein soll – stellen Sie sich diesen als Dreieck vor, das auf der einen Seite so dünn ist, dass dort keine Elektronen durchwandern können.**

**Liegt nun eine Spannung am Gate an, werden unterhalb des Isolators (hier orangefarben dargestellt) Elektronen angezogen (das Gate lässt sich je nach Ausführung mit positiver oder negativer Spannung betreiben). Der Kanal entsteht und ist negativ geladen; er wirkt wie eine Brücke und erlaubt den Stromfluss zwischen Source und Drain. Wäre die Gatespannung negativ, würde sie die Elektronen im Kanalbereich abstoßen und den Stromfluss in diesem Falle unterbinden.**

**Das Isolatormaterial bestand früher aus einem Metalloxid, üblicherweise Siliziumdioxid. Daher nennt man die Transistoren auch „Metal Insulator Semiconductor Field Effect Transistors“ oder einfach kurz: MOSFETs. Heute wird in der Regel dotiertes Polysilizium verwendet, die Bezeichnung MOSFET wurde aber beibehalten.**

# So rechnen Prozessoren binär (einfache Addition)

Bild: Intel

Bei den ersten Prozessoren konnte man die einzelnen Logikgatter noch erkennen; hier ein Ingenieur vor einer Rubylith-Folie, die zum Design früherer CPUs verwendet wurde

Prozessoren arbeiten nicht mit den von uns gewohnten Dezimalzahlen, sondern verwenden Binärzahlen. Diese sind wie im vorhergehenden Abschnitt beschrieben an die Eigenheiten der Elektronik angepasst und somit hervorragend für diesen Einsatzzweck geeignet.

Binärzahlen dienen dabei aber nicht nur zum Rechnen, sondern zur Darstellung jeglicher Form von Daten in einem Computer. Sie werden über das interne Bussystem weitergeleitet, im Speicher festgehalten oder über das Internet verschickt.

## Binäres Zählen

Dass Binärzahlen anders bezeichnet werden als Dezimalzahlen, sollte nicht abschrecken: Eine Neun ist für einen Prozessor ebenso eine Neun wie sie es für uns ist, aber er schreibt sie anders. Das hat technische Gründe: Prozessoren können nur zwischen zwei Zuständen unterscheiden. Entweder es sind elektrische Ladungen vorhanden oder nicht.

Zählen wir im Dezimalsystem hoch, so fangen wir bei 0 an, es folgt die 1, die 2 und schließlich kommen wir zur 9. Dann brauchen wir eine zweite Stelle, denn mehr Ziffern gibt es nicht. Also wird die 9 auf den Anfang zurückgesetzt (0) und die Ziffer links neben der 9 wird um eins erhöht. Links neben der 9 steht normalerweise keine Ziffer, deshalb nehmen wir einfach an, dass die Null dort steht, und erhöhen sie um eins. Und so folgt auf die (0)9 die 10. Diese Art der Zahlendarstellung heißt Stellenwertsystem und ist für uns heute vollkommen selbstverständlich.

Es gibt aber auch andere Systeme wie beispielsweise die römischen Zahlen: Hier ist die III kleiner als die V, obwohl sie aus mehr Ziffern besteht.

## Binäre Stellen

Auch Binärzahlen sind, wie für uns gewohnt, ein Stellenwertsystem. Der Unterschied ist, dass sie weniger Ziffern kennen; denn mit den beiden Zuständen „Strom“ und „Kein Strom“ lassen sich nur maximal zwei Ziffern realisieren. Diese werden üblicherweise als 0 und 1 bezeichnet. Zudem haben Prozessoren eine Maximallänge der Zahlen: Ein 32-Bit-Prozessor kann nur 32 Nullen oder Einsen aneinanderreihen, dann ist Schluss.

Für die meisten Beispiele hier verwenden wir jedoch einen 4-Bit-Prozessor: Das System ist dasselbe, doch bleibt die Rechnerei so einigermaßen übersichtlich.

## Binäre Addition (1)

Bevor es um den Teil des Prozessors geht, der zwei binäre Zahlen addiert, sollten wir diese Addition erst selbst verstanden haben. Die folgenden Erklärungen sind dann deutlich leichter zu verstehen. Die Rechenverfahren in Computern ähneln sehr stark den schriftlichen Verfahren, die Kinder in der Grundschule lernen: Sie sind schematisch und deshalb leicht anzuwenden, aus mancher Sicht aber wohl etwas umständlich. Kaum jemand würde an schriftliche Addition denken, wenn er 13 und 26 addieren möchte – und sobald die Zahlen zu groß werden, greift man ohnehin meistens zum Taschenrechner.

Möchten wir zwei binäre Zahlen addieren, müssen wir aber auf die schriftliche Addition zurückgrei-

fen. Das Vorgehen wird im folgenden kurz erklärt, ein bebildertes Beispiel findet sich auf der rechten Seite. Zunächst werden die beiden Zahlen übereinander geschrieben. Haben sie nicht dieselbe Länge, so muss die längere Zahl links hervorragen. Anschließend geht man die beiden Zahlen von rechts nach links durch und addiert immer zwei übereinanderliegende Ziffern.

Würden im Dezimalzahlensystem eine 3 und eine 4 übereinander stehen, so wäre das Ergebnis 7. Dieses könnten wir dann einfach darunter schreiben und eine Stelle weiter links weitermachen. Wären die beiden Ziffern aber beispielsweise 6 und 8, so wäre das Ergebnis 14. Hier müsste die 4 als Ergebnis aufgeschrieben werden und die 1 zusätzlich zu den beiden Ziffern eine Stelle weiter links addiert werden. Sie wäre ein sogenannter Übertrag (englisch „Carry“).

## Binäre Addition (2)

Im binären Zahlensystem ist das Vorgehen identisch, nur haben wir hier, wie bereits besprochen, nur zwei unterschiedliche Ziffern. Stehen zwei 0en übereinander, so ist das Ergebnis auch 0. Stehen eine 0 und eine 1 übereinander, so ist das Ergebnis 1. Stehen zwei 1en übereinander, ist das Ergebnis 0 und die Stelle eins weiter links erhält den Übertrag 1.

Als letzte Möglichkeit können noch drei 1en übereinander stehen: Je eine von den beiden zu addierenden Zahlen und vom Übertrag. Hier wäre das Ergebnis 1 mit erneutem Übertrag. All diese Varianten finden sich auch in unserem Beispiel. Dieses Schema der stellenweisen Addition wird für die gesamte Länge der beiden Zahlen durchgeführt und am Ende erhält man das Ergebnis. Ein Prozessor führt eine Addition auf exakt dieselbe Weise durch, wobei er sich dafür natürlich einer speziellen Schaltung bedient.

## Rechnen mit Strom

Zum binären Rechnen mit Strom brauchen wir nun „nur“ noch einen Schaltung, welche die binären Rechengesetze korrekt umsetzen kann. Wir müssen sicherstellen, dass zwei Nullen am Eingang eine Null am Ausgang ergibt. Ebenso müssen zwei verschiedene Spannungen am Eingang (0/1 bzw. 1/0) immer eine 1 am Ausgang ergeben und so weiter.

Um solche Gesetzmäßigkeiten abbilden zu können, gibt es sogenannte „Logikgatter“. Dabei handelt es sich letztlich um aus Transistoren gebaute Funktionseinheiten, die Eingangsströme in bestimmte Ausgangsströme umwandeln. Einige Beispiele solcher Schaltungen haben wir auf der folgenden Seite illustriert.

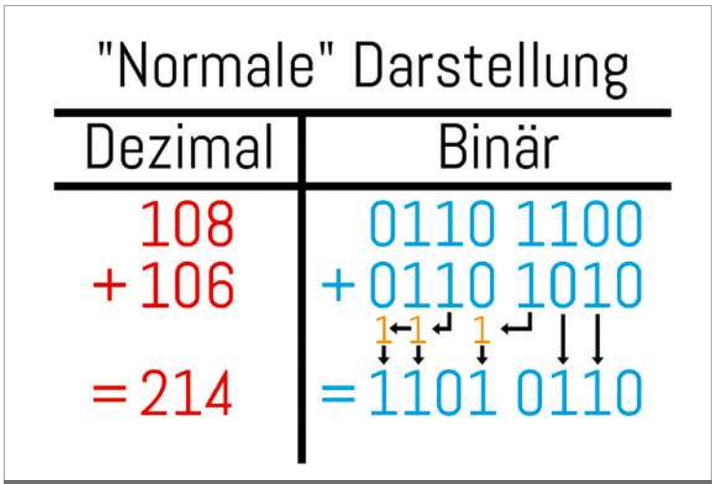
## Der Halbadder

Mit zwei solcher Logikgatter kann man bereits einen „Halbadder“ bauen, der die grundlegenden Rechenoperationen unserer Beispieladdition 108 plus 106, oder besser gesagt: 01101100 plus 01101010, korrekt umsetzt. Ein Halbadder ist einfach eine elektrische Schaltung, die aus zwei Ein-Bit-Ursprungswerten A und B ein Zwei-Bit-Ergebnis produziert, das aus einer Summe S und einem Carry-Bit („Übertrag“) C besteht. Der Aufbau ist im Schaubild rechts dargestellt.

## Gatter

Die für den Halbadder benötigten Funktionen („Gatter“) sind XOR („exklusives ODER“) sowie AND („UND“).

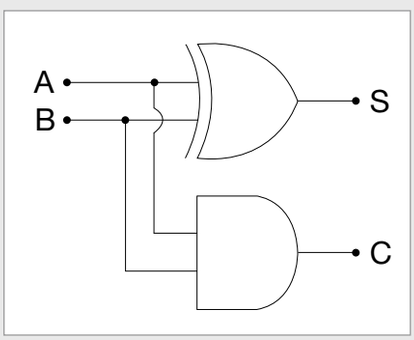
Für die eigentliche Rechenoperation ist das XOR-Gatter zuständig. Ein XOR liefert immer dann eine 1 als Ergebnis, wenn A und B voneinander verschieden sind, also eine 0 und eine 1 oder eine 1 und eine 0 sind. Sind beide gleich, also 0 und 0 oder 1 und 1, so ist das Ergebnis 0.




Einie Beispieladdition: Es sollen die binären Zahlen 01001100 (=dezimal 108) und 01101010 (dezimal 106) addiert werden. Die Binärzahlen sind hellblau dargestellt, Dezimalzahlen rot und der Übertrag orangefarben

## Wie ein Halbadder die Summe berechnet

**Halbadder: Das schematische Schaltbild zeigt die im Fließtext beschriebenen Operanden A, B, S und C sowie die logischen Operatoren XOR und UND. In der Abbildung werden Ein- und Ausgänge durch Linien symbolisiert, die einzelne Gatter verbinden. Kreuzen sich Linien, so gelten sie nur als verbunden, wenn sich an dieser Stelle ein Punkt befindet.**

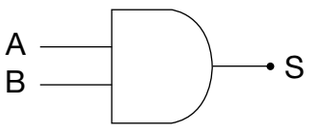



| A | B | C | S |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 0 |

## Übersicht: Grundlegende Logikgatter (Auswahl)

**Nachfolgend eine Übersicht häufig verwendeter Logikgatter; die Schaltzeichen links können je nach Land und Industrienorm unterschiedlich aussehen.**

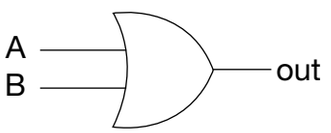
### UND-Gatter (AND)




| A | B | S |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

**Ein UND-Gatter gibt nur dann Strom aus, wenn an beiden Eingängen Strom anliegt**

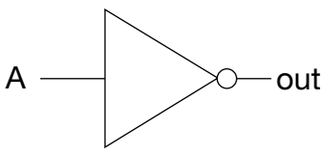
### ODER-Gatter (OR)




| A | B | S |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

**Anders das ODER-Gatter: Es liefert eine 1, wenn einer der beiden Eingängen eine 1 liefert – aber auch, wenn beide eine 1 eingeben!**

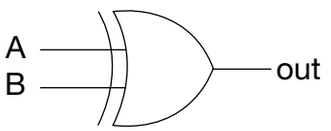
### NICHT-Gatter (NOT)




| A | S |
|---|---|
| 0 | 1 |
| 1 | 0 |

**Das NICHT-Gatter invertiert ein beliebiges Bit: Aus 0 wird 1, aus 1 wird 0**

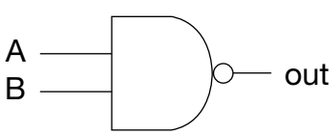
### Exklusives-ODER-Gatter (XOR)




| A | B | S |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

**Eine Schaltung, die viele beim ODER-Gatter erwarten würden: Ist einer der beiden Eingänge eine 1, ergibt der Ausgang ebenfalls eine 1 – andernfalls eine 0**

### Negiertes-UND-Gatter (NAND)




| A | B | S |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

**Das NAND-Gatter liefert eine 0, wenn beide Eingänge Strom führen – und ist somit ein invertiertes UND-Gatter**

ander differieren. Ansonsten liefert XOR eine 0 – perfekt! Denn genau das brauchen wir für die Addition der beiden Binärzählen, wenn wir wie ioben beschrieben jede Stelle einzeln addieren wollen. Zur Erinnerung: 0 und 0 gibt 0, wärend 0 und 1 eine 1 ergeben sollen und so weiter.

Es gibt nur ein kleines Problem: den Übertrag. In irgendeiner Form muss berücksichtig werden, ob die Eingangsoperanden 1 und 1 waren. Aus dem Ergebnis alleine können wir das nicht ablesen, denn 0 und 0 ergibt ebenfalls eine 0 am Ausgang.

Für diesen Zweck werden die Eingangsströme zusätzlich an ein zweites Gatter umgeleitet. Dieses Mal brauchen wir aber kein XOR-, sondern das oben bereits erwähnte UND-Gatter. Schließlich wollen wir nur wissen, ob beide Eingangsströme aktiv waren, ob wir also gerade 1 und 1 addieren oder nicht. Ist dies der Fall, soll am Ausgang ebenfalls Strom fließen, mithin eine 1 ausgegeben werden.

Das UND-Gatter liefert genau diese Information: Bei der Addition von 0 und 0 bleibt wie bei der Addition von 0 und 1 am UND-Gatter eine 0 am Ausgang stehen. Erst bei 1 und 1 am Eingang steht auch eine 1 am Ausgang.

### Wie die Summe korrekt berechnet wird

Mit dieser einfachen Schaltung können wir bereits rudimentär addieren – nicht schlecht! Doch leider reicht das noch nicht ganz für die Addition von 01101100 plus 01101010. Das Problem: Der Halbadder ignoriert alle vorherigen Überträge.

Um korrekt zu rechnen, brauchen wir aber genau diese Funktion. Eine Schaltung, welche die vorherigen Überträge berücksichtigt, heißt „Volladder“. Dieser hat drei Eingänge: einen für die aktuelle Ziffer der ersten Zahl, einen für die Ziffer der zweiten Zahl sowie einen dritten für einen etwaigen Übertrag. Ein Volladder ist nicht wesentlich komplexer, benötigt aber zusätzliche Logikgatter.

### Volladder

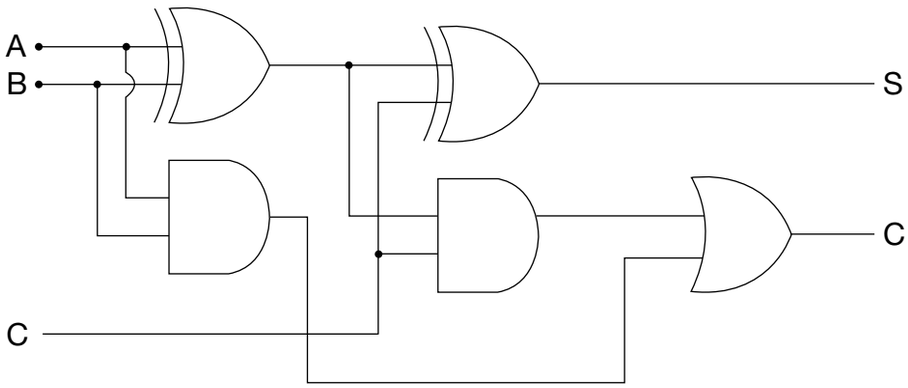
Konkret benötigen wir je ein weiteres UND- sowie XOR-Gatter. Zusätzlich brauchen wir ein neues ODER-Gatter. Beim ODER-Gatter wird eine 1 als Ergebnis geschrieben, wenn eine der beiden Ausgangszahlen eine 1 hatte. Wie genau er unseren Volladder komplettiert, sehen wir, wenn wir die Schaltung näher betrachten. Den schematischen Aufbau finden Sie auf der rechten Seite.

Der linke Teil des Volladders entspricht exakt dem bekannten Halbadder. Der Unterschied besteht darin, dass das Übertrags-Bit („Carry Bit“) im Anschluss der ersten Berechnung für eine zweite Berechnung genutzt wird. Dies entspricht exakt jenem Schritt, der auch beim manuellen Rechnen durchgeführt wird: Existiert ein Übertrag, so muss dieser mit der Summe der beiden Operanden verrechnet werden. In unserem Additionsbeispiel 01101100 plus 01101010 ist dies an der ersten, zweiten und vierten Stelle (von links) notwendig – sehen Sie sich hierzu bitte noch einmal die Abbildung auf der vorhergehenden Seite an, dort sind die Überträge mit einer orangefarbenen 1 kenntlich gemacht.

Das ist aber noch nicht alles. Es muss ein *neuer* Übertrag enstehen, wenn entweder der erste Rechenschritt (die Addition) oder der zweite Rechenschritt (die Verrechnung mit dem Übertrag) der Schaltung zwei 1er ausgibt. Dies wird mit dem neuen ODER-Gatter erreicht. Salopp gesagt, funktioniert es so: „Wenn

## Wie ein Volladder die Summe berechnet

**Der Volladder benötigt im Vergleich zum Halbadder drei weitere Logikgatter: XOR, UND sowie OR. In unseren Abbildungen werden Ein- und Ausgänge durch Linien symbolisiert, die einzelne Gatter verbinden. Kreuzen sich Linien, so gelten sie nur als verbunden, wenn sich an dieser Stelle ein Punkt befindet.**




| CI | A | B | Q | CO |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 |

eines der UND-Gatter positiv war (und damit beide Eingangswerte jenes Gatters eine 1 waren), dann gib den Übertrag einfach weiter an die nächste Stelle."

### Mehr Schaltungslogik

Sie haben nun im Wesentlichen die Schaltungslogik einer Addition kennengelernt. Unsere Beispiel-Schaltung könnte theoretisch beliebig lange Zahlen addieren – und das mit einem einzelnen Volladdierer und lediglich einer Handvoll Transistoren!

Der Nachteil allerdings ist, dass sie immer nur eine Stelle zweier Binärzahlen addieren kann. Jedes Mal muss das Ergebnis gespeichert und die nächsten Ziffern geladen werden. Das benötigt zusätzliche Zeit und eine einzelne Addition zweier Zahlen würde verhältnismäßig lange dauern. Schon bei einer 8-Bit-Zahl würden mindestens 24 Takte vergehen – für jede Stelle je ein Takt, um die Daten zu holen, die Berechnung durchzuführen und das Ergebnis zu schreiben (in der Praxis wären es noch viel mehr ...).

### Parallele Rechenwerke

Am einfachsten löst man dieses Problem mit einem parallel arbeitenden Addierer. Bei diesem hat man nicht einen einzelnen Volladdierer, sondern mehrere nebeneinander. Möchte man zwei Zahlen mit je acht Bit addieren, so verschaltet man einfach acht Volladdierer und das Ergebnis steht ohne erneuten Speicherzugriff direkt bereit.

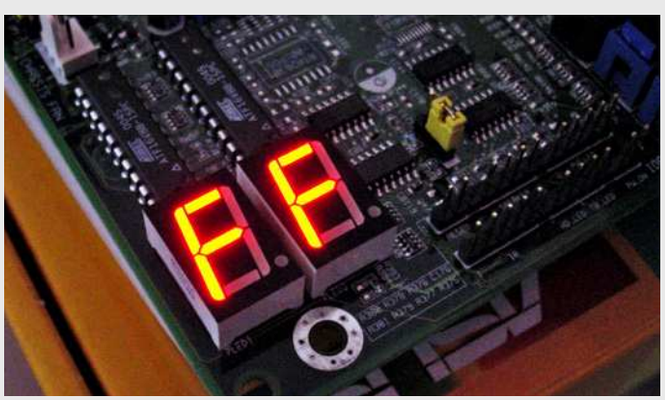
Ein Problem ist noch, dass der jeweils vorherige Volladdierer fertig gerechnet haben muss, damit der nachfolgende Volladdierer das richtige Carry-Bit erhält. Die Addition könnte so (vereinfacht dargestellt) in 10 Takten durchgeführt werden (Lesen, 8-mal Rechnen, Schreiben).

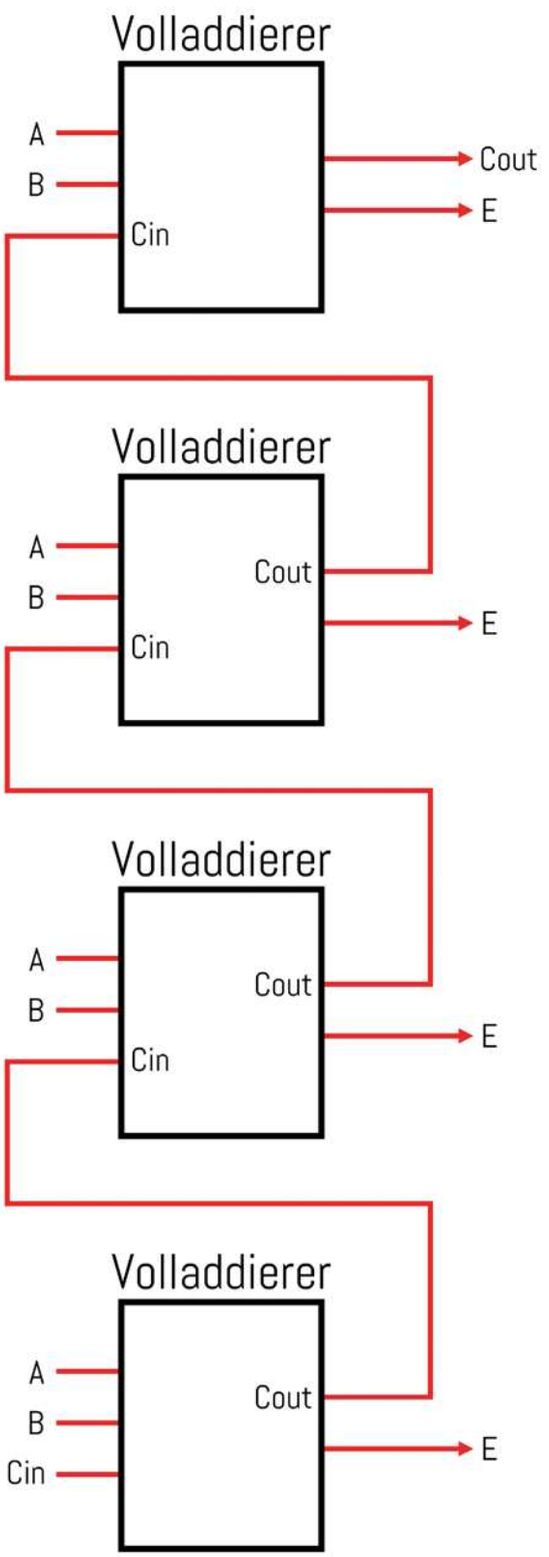
Aus diesem Grund gibt es besondere Aufbauten wie den Carry-Look-Ahead- oder den Carry-Skip-Addierer, die eine schnellere Berechnung des Carry-Bits ermöglichen. Hierbei

## Hexadezimalsystem

**Neben dem uns vertrauten Dezimalsystem und dem binären System gibt es noch weitere Zahlensysteme. Am bekanntesten dürfte das Hexadezimalsystem sein. Es verwendet als Basis 16 Ziffern – statt 2 wie beim binären oder 10 beim dezimalen System. Die Zahlen 0 bis 9 sind hier die gleichen wie beim Dezimalsystem. Um eine dezimale 10 darzustellen, verwendet es aber dann den Buchstaben A, für eine dezimale 11 das B bis hin zum F für die Dezimalzahl 15.**

**Für das dezimale 16 wird dann im Hexadezimalsystem eine zweite Stelle benötigt. Die Verwendung dieser Schreibweise hat praktische Gründe: Das binäre „0010 1101" lässt sich schlicht deutlich schlechter merken als das hexadezimale „2D". Gegenüber Dezimalzahlen hat man den Vorteil, dass man Hexadezimalzahlen leicht umwandeln kann: Immer vier Binärziffern ergeben genau eine Hexadezimalziffer, die rechten vier Ziffern der Beispielzahl ergeben 13 (dezimal) und somit D (hexadezimal); die linken vier Ziffern ergeben 2.**






Einzelne Volladdierer können beliebig oft hintereinander gehängt werden. Dadurch kann man Binärzahlen mit mehreren Stellen auf einmal addieren. Problematisch ist allerdings der Übertrag (Cin, Cout): Dieser muss alle Addierer durchwandern und verzögert dadurch die Berechnung

gilt es immer, einen sinnvollen Mittelweg zu finden: Eine Schaltung, die das Ergebnis zehn Prozent schneller bereitstellen kann, ist toll, es sei denn, sie benötigt die dreifache Menge Transistoren und damit deutlich mehr Strom.

## Mehr Gatter

Neben den drei vorgestellten Logikgattern gibt es eine Reihe weiterer, von denen wir die wichtigsten kurz vorstellen möchten.

Bei einer NICHT-Funktion wird die Zahl einfach invertiert, also überall, wo eine 0 stand, wird eine 1 ausgegeben und umgekehrt. Benötigt wird eine solche Umkehrung unter anderem für das Komplement (siehe folgende Seiten). Ähnliches gilt für das NAND-Gatter. Dieses entspricht letztlich nur einer UND-Schaltung mit nachgelagerter NICHT-Verschaltung. Sollen 0 und 0 am Eingang eine 1 am Ausgang erzeugen – und nicht eine 1 und eine 1 –, so kommt ein NAND-Gatter zum Einsatz.

Natürlich funktioniert das Ganze auch mit komplizierteren Zahlen. „NICHT 0111“ würde zum Beispiel als Ergebnis „1000“ liefern. Eine ODER-Verknüpfung zwischen „1001“ und „1010“ würde als Ergebnis „1011“ ausspucken, während ein XOR als Ergebnis „0011“ hätte. Dieselben beiden Zahlen mit UND statt mit ODER verknüpft, hätte „1000“ als Ergebnis.

## Mehr Rechenwerke

Analog zu den Varianten der Gatter gibt es mehrere Möglichkeiten, einen Volladder zu bauen. Die von uns vorgestellte Lösung wird häufig genannt, je nach Anforderungen lässt sie sich aber auch mit anderen Designs umsetzen.

Wie man sieht, kann man mit simplen logischen Funktionen binäre Zahlen manipulieren oder mathematische Berechnungen durchführen. Mit der richtigen Abfolge an Schritten ist ein simpler Prozessor bereits in der Lage, Zahlen zu addieren. Ein paar dieser Einheiten miteinander kombiniert – und man kann subtrahieren, multiplizieren oder Ganzzahlen dividieren.

Die Summe dieser Teilschaltungen wie Vergleicher, Halbadder, Volladder usw. ergibt das Rechenwerk in der Von-Neumann-Architektur. Dieses enthält eine oder mehrere ALUs (Arithmetic Logic Unit), welche die Berechnungen ausführen und gewissermaßen „Entscheidungen treffen“ können.

## Stromverbrauch und Kompromisse

Jedes Gatter hat einen bestimmten Stromverbrauch und eine Schaltzeit. Letztere gibt an, wie lange es braucht, bis das Gatter auf Änderungen am Eingang reagiert. Verbaut (und nutzt) man mehrere Gatter, so steigt auch der Stromverbrauch. Werden mehrere Gatter hintereinander verbaut, so steigt die resultierende Verzögerungszeit.

Diese beiden Eigenschaften sind bei der Entwicklung von Prozessoren ein Problem: Der Stromverbrauch soll möglichst niedrig und die Schaltzeit möglichst gering sein. Ersteres ist offensichtlich, letzteres hängt mit der Taktfrequenz zusammen. Diese kann maximal so hoch sein, dass selbst die langsamste beziehungsweise längste Kette von Gattern noch genügend Zeit hat, um zu reagieren.

Verschärft wird das Problem, je länger die interne Zahlenlänge ist. Was bei 32-Bit-Zahlen noch in akzeptabler Zeit funktioniert, kann bei 64 Bit langen Zahlen schon zu inakzeptablem Verzögerungen führen (mehr zu Registern im weiteren Verlauf).

# So rechnen Prozessoren (komplexe Operationen)

Wir wollen nicht zu tief in die Binärarithmetik und die Logik der Gatter einsteigen – doch ein wichtiges Problem muss noch erwähnt werden ...

## Negative Zahlen

Die Binärzahlen, wie wir sie bis jetzt kennen, können zwar Zahlen darstellen, aber nur ganze positive Zahlen und die Null. Wie stellt man aber binär beispielsweise eine dezimale -4 dar?

Um negative Zahlen darzustellen, kann man ein zusätzliches Bit hinzufügen, ein sogenanntes „Vorzeichen-Bit“. Dieses würde in der x86-Welt links „angehängt“. Ist es 0, so ist die Zahl positiv, ist es 1, ist die Zahl negativ. Allerdings kann man mit dieser einfachen Lösung nicht immer korrekt rechnen.

Als Beispiel soll die dezimale Addition von 3 und -4 dienen. Das richtige Ergebnis ist -1. Bei einer binären Darstellung würden 0011 (dezimal 3) sowie 1100 (dezimal -4, die ganz linke Stelle gibt das Vorzeichen an) verrechnet, was 1111 und damit nach unserer Logik -7 (dezimal) ergeben würde.

## Einerkomplement

Abhilfe schaffen kann das sogenannte „Einerkomplement“. Beim Einerkomplement werden alle Stellen einer Binärzahl invertiert. Aus 1011 wird demnach 0100. In Kombination mit dem erwähnten Vorzeichenbit kann man nun korrekt rechnen! Wieder unser Beispiel: Die Addition von 3 und -4 stellt sich binär nun wie folgt dar: 0011 (+3, unverändert) und 1011 (-4, komplementär) ergeben 1110. Das Ergebnis 1110 beginnt mit einer 1, ist also negativ. Bilden wir von den verbleibenden Stellen das Komplement, ergibt dieses 001, also eine dezimale 1 – die aufgrund des Vorzeichenbits als -1 interpretiert wird.

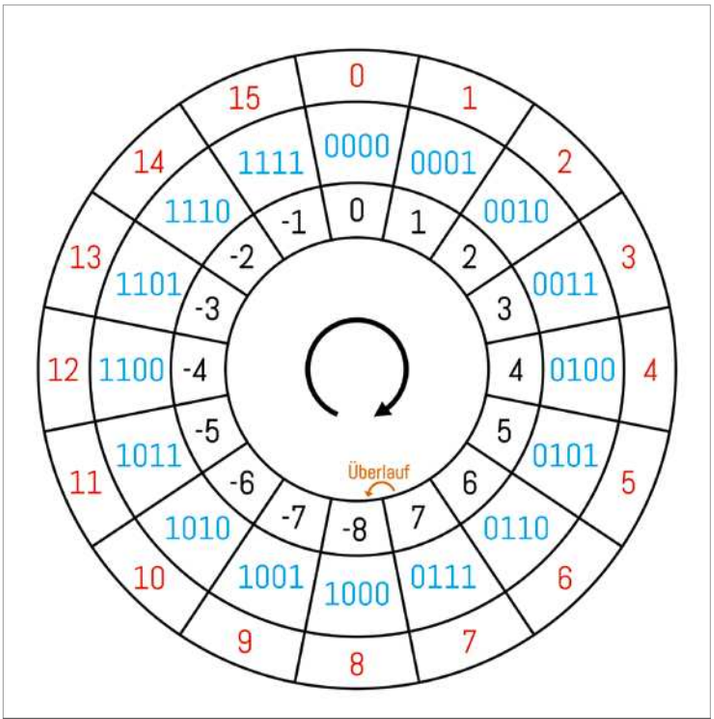
Doch auch damit ist das Problem noch nicht vollständig gelöst. Zwar gibt es nun auch negative Zahlen, aber auch eine negative Null: Gehen wir von einem Byte aus, so würde 0000 0000 der Zahl +0 entsprechen und 1000 0000 der -0. Aus mathematischer Sicht ist das unsinnig, deshalb wurde das sogenannte Zweierkomplement entwickelt.

## Das Zweierkomplement

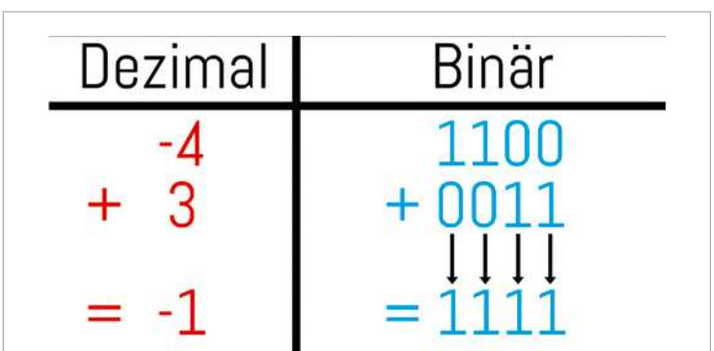
Das Zweierkomplement funktioniert exakt wie das Einerkomplement – mit dem Unterschied, dass am Ende jeweils noch eine binäre 1 hinzuaddiert wird. Aus der dezimalen -4 wird jetzt die binäre Ziffer 1100 (das Komplement von 4 ist 1011 plus 1 ergibt 1100; der Übertrag im letzten Schritt wird ignoriert). Wichtig: Das Zweierkomplement wird wie das Einerkomplement nur für die negative Zahl gebildet, die dezimale 3 bleibt binär unverändert 0011. Führen wir nun die binäre Rechnung durch, lautet das Ergebis 1100 plus 0011, was 1111 ergibt. Das Ergebnis müssen wir nach den Zweierkomplement-Regeln wieder zurückwandeln, was am Ende -1 ergibt (nehmen Sie am besten die kreisförmige Komplement-Darstellung oben rechts zu Hilfe).

## Interpretationssache

Mit dem Zweierkomplement ändert sich also die „Interpretation“ eines Bytes. Man kann mit einem Byte nicht mehr die Zahlen von 0 bis 255 darstellen, sondern die Zahlen von -128 bis 127. Diese Darstellung der Zahlen ist vor allem deshalb vorteilhaft und gewissermaßen al-




Das Zweierkomplement in der Übersicht: Binärzahlen sind hellblau, Dezimalzahlen in klassischer Darstellung rot (äußerer Kreis). Die komplementär dargestellte Dezimalzahl ist im innersten Kreis abgebildet. Die vier Bits (im Fachjargon: Nibble) stehen damit nicht mehr für die Zahlen von 0 bis 15, sondern für die Zahlen -8 bis 7. Das erste Bit (ganz links) zeigt das Vorzeichen




Das Zweierkomplement „in Aktion“: Die dezimale, positive 4 ist binär geschrieben 0100. Ihr Zweierkomplement lautet 1100 (siehe Fließtext). Bei der dezimalen 3 ändert sich nichts. Das binäre Rechenergebnis 1111 ergibt in der Zweierkomplement-Darstellung den richtigen Wert (dezimal -1)

ternativlos, weil sie ohne weitere Änderungen mit den vorher gezeigten Volladdierern funktioniert: Man kann einfach zwei beliebige positive oder negative Zahlen addieren und das Ergebnis ist (fast) immer richtig.

## Überlauf

Eine Veränderung gibt es beim Überlauf, den wir bisher noch nicht besprochen haben: Hat man eine Binärzahl, die nur aus 1en besteht, im Falle eines Bytes also 1111 1111, und addiert man 1 hinzu, so würde 1 0000 0000 als Ergebnis herauskommen. Da die neunte Stelle aber nicht dargestellt werden kann, kommt es zu einem Überlauf. Eine Binärzahl in klassischer Darstellung würde dadurch auf Null zurückgesetzt werden, die neunte Ziffer fällt weg. In der Komplement-Darstellung gibt es dasselbe Problem, aber bei der größten positiven Zahl. Addiert man zu einer 0111 1111 (127) eine 1 hinzu, so wird eine 1000 000 (-128) daraus.

Bei älteren Prozessoren wie dem MOS 6569, der im legendären C64 verbaut wurde, kann man Rundungsfehler bei Gleitkommaberechnungen „live“ beobachten. Eine Demo gibt‘s unter vimeo.com/330159336

Binäre Zahl ohne Kommastelle

| 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 |
|---|---|---|---|---|---|---|---|
| 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 |

Ergebnis: 1x128 + 1x32 + 1x4 + 1x2 = 166

Binäre Zahl mit Kommastelle

| 16 | 8 | 4 | 2 | 1 | 0,5 | 0,25 | 0,125 |
|---|---|---|---|---|---|---|---|
| 1 | 1 | 0 | 0 | 1 | 0 | 1 | 0 |

Ergebnis: 1x16 + 1x8 + 1x1 + 1x0,25 = 25,25

Umrechnung von binären Zahlen mit und ohne Kommastelle ins Dezimalsystem: Das Vorgehen ist dabei gleich wie bei der„normalen“ Umrechnung. Für jede 1 wird die Zahl der jeweiligen Spalte hinzuaddiert. Zu bedenken ist, dass der Trennpunkt zwischen der 0,5 und der 1 an beliebiger Stelle gesetzt werden kann – je nach Interpretation

Überläufe sind dabei kein theoretischer Sonderfall, sondern kommen tatsächlich vor: Sofern der Programmierer nicht verhindert, dass ein solches Szenario eintreten kann, können auch moderne Prozessoren zu falschen Ergebnissen kommen. Beliebt sind solche Fehler vor allem bei Zahlen, denen wenig Beachtung geschenkt wird – beispielsweise bei Ingame-Statistiken. Stehen dort große negative Zahlen, wo eigentlich positive Zahlen zu erwarten gewesen wären, so hat der Programmierer einen Fehler gemacht und der Spieler die Zahl zum Überlaufen gebracht.

## Weitere Rechenarten

Da unser Prozessor jetzt positive und negative Zahlen darstellen und addieren kann, kann er auch alle üblichen Rechenarten durchführen: Die Subtraktion wird zur Addition mit einer negativen Zahl, die Multiplikation zur mehrfachen Addition und die Division zur mehrfachen Addition mit einer negativen Zahl.

Probleme gibt es einzig beim letztgenannten Verfahren, denn teilt man zwei Zahlen durch einander, so bleibt manchmal ein Rest übrig. Zudem können wir im Moment noch keine Nachkommastellen darstellen, doch zumindest den Teiler und den Rest können wir bereits herausfinden. Auch das ist wieder praxisrelevant: Programmierer müssen sich entscheiden, ob sie Kommazahlen oder ganze Zahlen verwenden. Ersteres ist langsamer, Letzteres weniger flexibel.

## Darstellung von (Gleit-) Kommazahlen

Für die Darstellung von Kommazahlen gibt es eine simple Lösung: Wir denken uns einfach einen Punkt (bzw. ein Komma) in unserer binären Zahl. Links des Punktes steigen die Stellenwerte wie gewohnt an, rechts davon sinken sie schrittweise und werden so immer kleiner. Da wir im Binärsystem arbeiten, zeigt die erste Stelle nach dem Komma aber nicht den zehnten Teil einer ganzen Zahl, sondern jeweils die Hälfte. Beispielsweise kommt rechts neben der 1 die ½ = 0,5, gefolgt von der ¼ = 0,25. Auch mit dem gedachten Punkt funktioniert unser Addierer weiterhin, es sind keine Anpassungen notwendig.

Ein Problem ist allerdings die Genauigkeit, denn wir können so nur manche Kommazahlen darstellen. Haben wir viele Stellen nach unserem „Trennpunkt“ so werden diese Kommazahlen genauer, aber die Lösung ist noch nicht optimal.

## FPU

An diesem Punkt verlassen wir erstmals die ALU und wechseln zur FPU, die mit sogenannten Gleitkommazahlen arbeitet. Diese haben, wie es der Name bereits andeutet, den Trennpunkt nicht an einer festen Position.

Stattdessen gibt es für Binärzahlen bestimmter Längen Standards, die die Aufteilung der Zahl angeben. Das erste Bit gibt weiterhin das Vorzeichen an, dann folgen mehrere Bits, die die Position des Punktes angeben, und zuletzt noch die eigentliche Zahl. Dadurch kann man sehr flexibel große und kleine Zahlen darstellen, indem man einfach den Trennpunkt verschiebt. Auf die exakte Berechnung solcher Zahlen verzichten wir an dieser Stelle, Gleitkommazahlen von Hand zu addieren, benötigt mehrere Schritte und wäre zu kompliziert.

## Single- und Double-Precision

Wir müssen uns Gleitkommazahlen aber gar nicht näher ansehen, um zu verstehen, wieso es in CPUs und GPUs verschiedene Arten von ihnen gibt: zumeist Single- und Doub-

le-Precision, es existieren aber auch Half- und Quad-Precision.

Für viele Anwendungen reichen Gleitkommazahlen mit 32 Bit Gesamtlänge – Single-Precision – vollkommen aus. Für möglichst genaue Ergebnisse, wie beispielsweise bei wissenschaftlichen Berechnungen, muss aber auf Double- (64 Bit) und Quad-Precision (128 Bit) gewechselt werden. Im Umkehrschluss kann man die Genauigkeit mit der Verwendung von Half Precision (16 Bit) auch reduzieren.

Aus Sicht des Programmierers hat eine höhere Genauigkeit nur Vorteile, sie beansprucht die Hardware aber deutlich stärker: Möchte man mit 64- statt mit 32-Bit-Gleitkommazahlen rechnen, so muss die Schaltung eben doppelt so viele Ein- und Ausgänge haben und ist damit doppelt so komplex.

## Genauigkeit gegen Geschwindigkeit

Aus diesem Grund haben Grafikkarten hauptsächlich Einheiten für Single-Precision-Berechnungen: Das ist der sinnvolle Mittelweg. Zwar gibt es auch Schaltungen, die sich kombinieren lassen, sodass zwei 32-Bit-Gleitkommaeinheiten eine 64-Bit-Zahl bearbeiten können, doch sind diese im Vergleich zu spezialisierten Einheiten aufwendiger. Sie benötigen mehr Bauteile und mehr Strom.

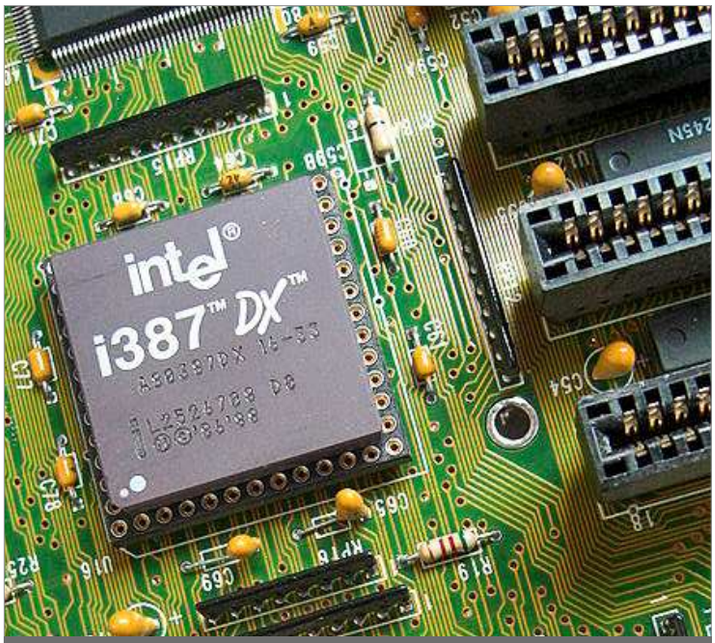
## Rundungsfehler

Bei der Weiterverarbeitung von Gleitkommazahlen sollte man stets im Hinterkopf behalten, dass eine solche Zahl lediglich eine mehr oder minder genaue Rundung darstellt. Je geringer dabei die Anzahl an Bits, die für die Mantisse zu Verfügung stehen, desto ungenauer die Rundung. Ein Beispiel: Nehmen wir die Berechnung 10 geteilt durch 3. Dabei kommt 3,3 Periode heraus. Mit nur einer Nachkommastelle erhält man das Ergebnis 3,3. Multipliziert man dieses Ergebnis mit 1.000, kommt man auf 3.300 – was falsch ist! In Wahrheit ist das Ergebnis 3.333,3 Periode.

## Designfragen

Die beschriebenen Einheiten zum Verrechnen binärer Zahlen sind nur ein kleiner Teil des Rechenwerkes in Prozessoren. Es werden noch viele weitere Bestandteile benötigt: Ein Prozessor muss auch Zahlen vergleichen und Programmdaten laden können, nur dann ist er imstande, Programme auszuführen.

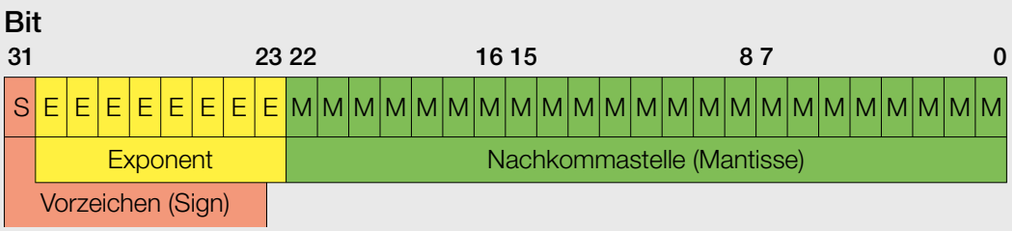
Dabei müssen die Ingenieure in der Chipentwicklung immer Kompromisse schließen – auch heute noch. Man kann wenige Einheiten verbauen, was kostengünstig, aber dafür langsam ist, oder man verbaut komplexe Schaltungen, die Aufgaben auf einmal abarbeiten können. Ähnliches gilt für spezialisierte Einheiten, die schnell sind, aber selten genutzt werden. Ist da eine häufiger genutzte, aber langsamere Einheit die bessere Wahl? Das muss wohl für jede CPU neu entschieden werden.




Zu Zeiten der 386er-Generation war es üblich, Co-Prozessoren nachzurüsten. Dafür gab es eigens einen Sockel auf dem Mainboard. Mit dem 486 begann später die in die CPU.

## Wie eine Gleitkommazahl zusammengesetzt ist

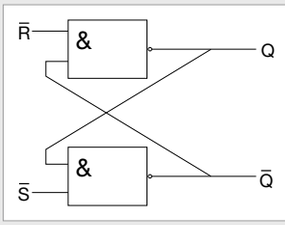
**Eine Gleitkommazahl besteht grundsätzlich aus drei Elementen: der Mantisse, der Basis und einem Exponenten. Die Zahl 17,5 dargestellt als Gleitkommazahl würde man 1,75 mal 10 hoch 1 schreiben. Um eine solche Zahl in ein 32 Bit breites Register zu quetschen, muss man zuerst einmal definieren, welche Stellen wofür verwendet werden sollen. Gemäß Spezifikation IEEE 754 für Float-Zahlen mit einfacher Genauigkeit wird dabei ein Bit für das Vorzeichen verwendet (eine Gleitkommazahl kann ja auch negativ sein), 8 Bit für den Exponenten und der Rest für die Nachkommastellen (Mantisse).**



# So funktionieren Register und Flip-Flops

## Wie ein Flip-Flop arbeitet

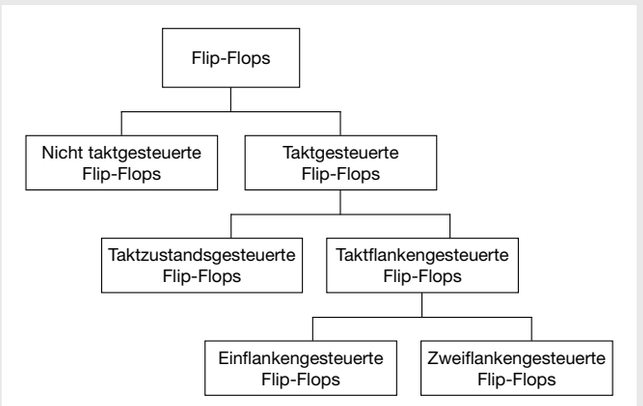
**Das Flip-Flop stellt die einfachste Form eines Speichers dar und kann ein Bit so lange speichern, wie Strom anliegt (sogenannter flüchtiger Speicher).**




| R | S | Q |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 0 |
| 1 | 1 | 0 |

## Flip-Flop-Typen

**Es gibt nicht den einen Flip-Flop-Speicher; stattdessen wurden im Laufe der Zeit viele verschiedene Typen entwickelt, die alle für unterschiedliche Anforderungen optimiert sind. In Prozessoren kommen meist flankengesteuerte Flip-Flops zum Einsatz; diese behalten ihre Information auch dann, wenn sich das Eingangssignal ändert.**




Neben den Gattern gibt es weitere aus Transistoren aufgebaute Schaltungen, die für den Betrieb eines Rechenwerks wichtig sind.

### Flip-Flops

An vorderster Stelle steht das sogenannte Flip-Flop, welches einen Zustand (1 oder 0) über einen längeren Zeitraum speichern kann. In der einfachsten Ausführung, einem ungetakteten RS-Flip-Flop, werden zwei NAND-Gatter (engl. für „Nicht-Und"-Gatter) kombiniert, die zwei Eingänge R und S schalten zwei Ausgänge Q und ¬Q (siehe Grafik links).

### Register

Aus Flip-Flops bestehen auch die prozessorinternen Speicher, die sogenannten Register. Diese beseitigen einen der vielen Flaschenhälse in der ursprünglichen Von-Neumann-Architektur. Register sind schnelle Zwischenspeicher direkt im Kern des Prozessors. Damit stellt der Prozessor letztlich seine eigentlichen Berechnungen an. So müssen die Daten nicht im langsamen Arbeitsspeicher miteinander verrechnet werden.

### Registersätze

Ein klassischer x86-PC-Prozessor verfügt über 14 Register: vier bzw. acht Universalregister (GPR, General Purpose Register, dazu gleich mehr) sowie zehn spezialisierte Register, in denen Zahlen zur Berechnung abgelegt werden oder Zeiger auf die Stelle des Programmcodes gesetzt werden können, an der sich der Rechner gerade im Programmablauf befindet. Ein Prozessor kann Operationen auf Länge der Register durchführen. Das bedeutet, dass ein 32-Bit-Prozessor Operationen mit 32-Bit-Zahlen durchführen kann, also mit Zahlen, die 32 Nullen oder Einsen enthalten können. Jeder Operand bei mathematischen Operationen liegt also im Bereich von 0 bis 4.294.967.295 (= 2 hoch 32 Möglichkeiten). Die Größe eines Operanden, das heißt die Anzahl der Binärstellen, die in den Registern verarbeitet werden kann, wird auch als Wortlänge oder Wortbreite bezeichnet.

### Arbeiten mit Registern

Der normale Anwendungsprogrammierer, der eine Programmierhochsprache wie C++, Visual Basic oder Java beherrscht, kommt normalerweise mit Registern nie in Berührung. Er programmiert mit einigermaßen verständlichen Prozeduren, Funktionen, Klassen und Bibliotheken. Anders sieht die Sache allerdings auf Maschinensprache-Ebene aus. Hier können die Register eines Prozessors unmittelbar beschrieben, ausgelesen und miteinander verarbeitet werden. Im Endeffekt werden natürlich auch die Quellcodes von Hochsprachen in Maschinensprache übersetzt. Allerdings übernimmt diese Arbeit der Compiler, während der Assembler-Programmierer diese Abstraktion nicht zur Verfügung hat.

Doch egal ob Hochsprachen- oder Assembler-Compiler die Aufgabe übernehmen, die vier Register AX, BX, CX sowie DX können in einem 8086-Prozessor, dem Urkeim aller aktuellen PC-Prozessoren, für universelle Operationen verwendet werden. Als Folge dessen müssen alle Nachfolger dieses Mikroprozessors diese Register besitzen und deren Verwendung identisch handhaben.

### 16-Bitter im 8-Bit-Pelz

Nun wäre selbst der einfachste x86-Prozessor in seiner Leistung

extrem begrenzt gewesen, hätte er nur mit 8-bittigen Zahlen arbeiten können – schon bei Werten größer als 256 ($2^8$) wären dramatische Leistungseinbußen bei der Berechnung die Folge gewesen.

Die CPU-Architekten haben den Ur-Prozessoren daher von Anfang an 16 Bit breite Register mitgegeben, die Register AX, BX, CX und DX sind also „eigentlich" nicht vier, sondern acht Register zu jeweils acht Bit. Tatsächlich lassen sich diese einzeln ansprechen und übernehmen bei 16-Bit-Operationen unterschiedliche Aufgaben: Der eine Teil speichert die höherwertigen Bits („High", AH), der andere die niederwertigen Stellen („Low", AL).

## Spezialregister

In der Übersicht auf der rechten Seite oben kann man die Funktionsteilung sehr gut erkennen, ebenso wie die übrigen der insgesamt 14 Register. Diese „übrigen" sind Spezialregister und nur für bestimmte Zwecke vorgesehen. Sie dienen dem internen Programmablauf und sind ebenfalls 16 Bit breit. Bis auf eine Ausnahme können diese aber nicht in zwei Teile untergliedert bzw. separat angesprochen werden. Die Spezialregister beschreiben wir in den folgenden Abschnitten ausführlich.

## Daten- und Arithmetikregister

Neben den Registern AX, BX, CX und DX sind vier weitere für den Programmablauf entscheidend: diese Zellen werden als SP/BP und SI/DI bezeichnet.

Die Register SP und BP verwalten den sogenannten „Stack" (engl. „Stack" = Stapel oder Speicher). Der Stack dient unter anderem dazu, Rücksprungadressen beim Aufruf von Unterprogrammen zu speichern, damit der Prozessor nach dem Abarbeiten des Unterprogramms seine Arbeit im übergeordneten Programm fortsetzen kann. Der Stack arbeitet nach dem LIFO-Prinzip (Last In First Out), eben wie ein echter Stapel. Was man zuletzt oben aufgelegt hat, muss man auch zuerst wieder entfernen (das „Herausziehen" eines Wertes ist nicht vorgesehen). In das 16-Bit-Register SP würde ja nur eine einzelne Zahl passen, noch dazu eine begrenzte ($2^{16}$ = 65.536); folglich speichert der Prozessor in diesem Register lediglich eine Speicheradresse, und zwar jene, an der der jüngste Stapelwert zu finden ist – eben der, der zuletzt „aufgelegt" wurde.

Der Base Pointer dient dazu, einen bestimmten Platz im Stapel zu speichern – ebenfalls in Form einer Speicheradresse. Für die Programmierer unter den Lesern: Über diese Hintertür lassen sich Funktionen relativ komfortabel realisieren: Beim Aufruf einer Funktion wird die Stapeladresse im Base Pointer gesichert, beim Verlassen von dort wieder zurück auf den Stapel gelegt.

SI und DI steht für „Source Index" sowie „Destination Index", also Quell- und Zielregister. Hier geht es vor allem um Operationen, die Zeichenketten verarbeiten – zum Beispiel, wenn ein Suchbgeriff innerhalb eines Textes gefunden werden soll. Das Register SI speichert in einem solchen Fall die Adresse, an welcher der zu durchsuchende Speicherbereich beginnt, DI analog dazu die entsprechende End-Adresse.

## Programmzähler/ Befehlszähler

Der Programmzähler („Instruction Pointer", IP) enthält die Speicheradresse des nächsten einzulesenden Programmteils. Bei einem sequenziellen Abarbeiten des Programmcodes zählt dieser einfach hoch.

## Statusregister (Flag-Register)

Nach manchen Operationen kann es vorkommen, dass zusätzliche Informationen für die nachfolgende Operation bereitgestellt werden müssen. Dies geschieht, indem verschiedene Bits im Statusregister, die Status-Flags, gesetzt werden (im Schema links als „FLAGS" bezeichnet).

Zu den Flags gehört beispielsweise das Zero-Flag. Dieses wird immer dann gesetzt, wenn das letzte Ergebnis gleich null ist. Hierdurch lassen sich mit geringem Aufwand beispielsweise Schleifenbedingungen realisieren. Ein weiteres Flag ist das Carry-Flag, welches für den Übertrag benötigt wird, wie wir im vorhergehenden Abschnitt gesehen haben. Dies muss bei möglichen Nachfolgeoperationen in der Regel berücksichtigt werden.

Das Overflow-Flag zeigt beim Rechnen mit vorzeichenbehafteten Werten an, ob es einen Übertrag in das Vorzeichenbit gab, wodurch die Zahl verfälscht würde. Ob eine Zahl vorzeichenbehaftet ist, zeigt das Sign-Flag an. Darüber hinaus gibt es noch viele andere Flags, die häufig für spezielle Anwendungszwecke eingesetzt werden.

Insgesamt beherbergt dieses Register 12 Flags; da es wie dies anderen Register 16 Bit breit ist, bleiben vier Bits (12-15) ungenutzt, also „leer".

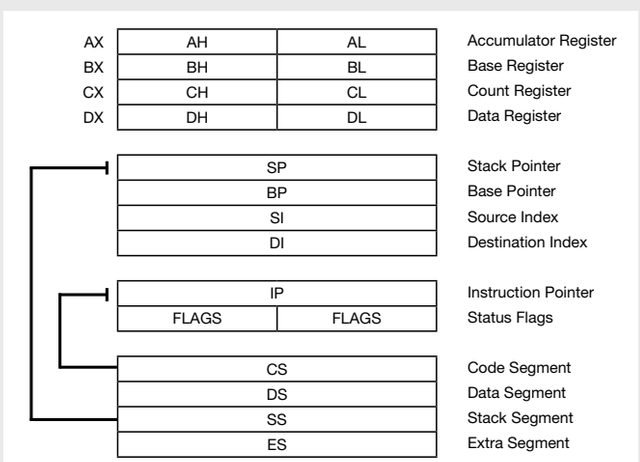
## Adressregister

Es bleiben noch vier Register übrig, im Schaubild auf der vorhergehenden Seite ganz unten zu finden. Diese Register sind im Grunde das Ergebnis der Verrenkungen, die Intel machen musste, um in den Anfangstagen der x86-Ära mehr Speicher nutzen zu können.

Weil mit 16 Bit breiten Registern maximal $2^{16}$ Bytes (ca. 64 KiByte) angesprochen werden können, nutzen die CPU-Entwickler eine andere Methode, um gesuchte Bytes im RAM zu finden. Jede Speicheradresse bestand einfach aus *zwei* 16-Bit-Werten; der eine bezeichnete ein Speichersegment, der anderen einen

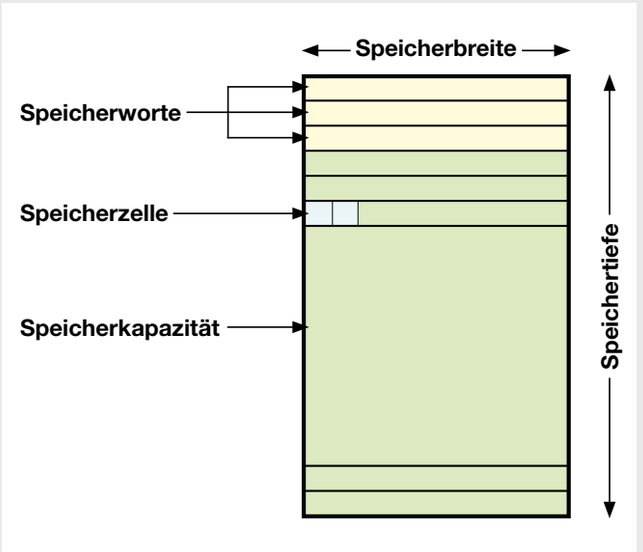
## Registersatz einer x86-CPU

**Diese Register besitzt ein klassischer 8086-16-Bit-Prozessor. Bei 32-Bit-CPUs heißen die verlängerten Register EAX, EBX, usw. („E" für „Extended")**



# Speicherorganisation

**Der Hauptspeicher ist zeilenweise organisiert. Eine Zeile bildet ein Speicherwort und hat eine feste Speicheradresse. Jede Speicherzelle ist ein Bit groß.**




# Struktur eines Maschinenbefehls

**Programme liegen dem Prozessor in Form von Maschinencode im Binärzahlenformat vor. Da ein solcher Code aber äußerst kryptisch und praktisch nicht lesbar ist, erfolgt die Programmerstellung häufig in Form einer Hochsprache, beispielsweise in C. Bei hardwarenaher Programmierung kann der Code auch direkt in prozessorspezifischer Assemblersprache geschrieben werden. Hierbei handelt es sich quasi um Maschinensprache, die allerdings mit anschaulichen Kürzeln, sogenannten Mnemonics, lesbar gemacht wird (siehe Seite 11). Der fertige Code wird anschließend in Binär-Maschinensprache übersetzt.**

**Ein Maschinenbefehl besteht aus mehreren Teilen. Den ersten bildet der OpCode, der angibt, um welchen Befehl es sich handelt, beispielsweise eine Addition, das Laden eines Wertes in ein Register oder einen Sprungbefehl. Die Gesamtheit der für den Prozessor verfügbaren Befehle bildet den Befehlssatz. Auf den OpCode folgen Angaben zu den Operanden und deren Adressierungsart. Je nach Befehl wird zudem noch angegeben, in welches Register bzw. an welche Speicheradresse das Ergebnis gespeichert werden soll.**

```
load R1,A
load R2,B
add R3,R1,R2
store C,R3
```

So könnte die Operation C = A + B in Pseudo-Assemblercode aussehen. Die Werte A und B werden zunächst in separate Register geladen, dann addiert und in einem dritten Register gespeichert. Danach wird der Registerwert im Wert C gespeichert.

| OpCode | Zielregister | Quellregister 1 | Quellregister 2 |
|---|---|---|---|
| add | R3 | R1 | R2 |

Offset (dt.: „Versatz“). Grob veranschaulicht, könnte man Segmente mit Straßennamen gleichsetzen und den Offset mit einer Hausnummer. Der Offset gibt dabei immer an, wie viel „weiter“ man gehen muss, um zur gesuchten Adresse zu gelangen.

## Segmente und Offset

Wir wollen nicht zu weit in die Tiefe gehen, wichtig ist vor allem, dass man für diese Teilung in Segment und Offset zusätzliche Register brauchte, welche die Adresse für das jeweilige Segment enthalten mussten. Das Register CS speichert diese Zahl für den gerade aktuellen Code („Code Segment“), das Register DS macht das gleiche für das jeweils aktuelle Segment des Datenbereichs („Data Segment“). Für den Offset gibt es kein eigenes Register, denn dieser Wert wird in die „normalen“ Register geladen.

„SS“ und „ES“ schließlich speichern ebenfalls Segment-Adressen: SS jene, die für Adressen auf dem Stack benötigt werden („Stack Segment“), ES enthält das Segment einer Speicheradresse, die für Stringoperationen genutzt wird („Extra Segment“) – was diese beschleunigt.

## Von 16 zu 32 Bit

Das Bearbeiten größerer Zahlen mit16-Bit breiten Registern ist ziemlich umständlich, da die Zahlen dann in zwei oder mehrere Teile aufgespaltet werden müssen. Schnell wurden 16-Bit-Prozessoren daher von 32-Bit-CPUs abgelöst. Intels erster 32-Bit-x86-Prozessor kam mit dem 386 schon 1985 auf den Markt.

Doch auch diese Architekturvariante war noch lange nicht perfekt. Ein 32-Bit-Register kann auch nur eine beschränkte Anzahl an Adressen im Arbeitsspeicher aufnehmen; für ein 32-Bit-System bedeutet das: Bei 4 GByte RAM ist Schluss. Mehr Speicher kann ein 32-Bit-Prozessor nicht adressieren (zumindest nicht auf herkömmlichem Wege).

## Von 32 zu 64 Bit

Die Einführung von 64-Bit-Prozessoren, also von Prozessoren mit 64 Bit breiten Registern, entschärfte das Problem, indem hier rein rechnerisch 2 hoch 64 Byte RAM adressiert werden können, also satte 16.777.216 Terabyte. Eine schier unvorstellbar große Menge. Aus diesem Grund haben die aktuellen 64-Bit-x86-Prozessoren eine freiwillige Selbstbeschränkung: Nur 40 Bit stehen für die Adressierung des Arbeitsspeichers zur Verfügung. Das sind 1 TByte RAM, was für eine Weile genügen sollte.

Wichtig am Ende unseres kleinen Registerausfluges ist noch ein Hinweis: Alle diese Angaben gelten nur für Intels x86-Architektur (und kompatible); andere Architekturen können gänzlich anders aufgebaut sein und mit unterschiedlichen Registern arbeiten.

## FPU-Einheit

ALUs und Register sind elementare Bauteile eines x86-PCs. Wie wir im Abschnitt über das binäre Rechnen gesehen haben, sind es aber vor allem die Gleitkommaoperationen, die aufwendig sind und viel Zeit benötigen. Die logische Konsequenz in der CPU-Entwicklung war eine eigene Gleitkommaeinheit, englisch: „Floating-Point-Unit“ oder FPU.

Aus einem modernen PC-Prozessor ist dieses Bauteil nicht mehr wegzudenken. Elementar für eine x86-CPU ist es allerdings nicht. Weder die ersten 8086- und 8088-Prozessoren besaßen eine FPU, noch die 80286- und 80386-Prozessoren. Erst der 80486-Prozessor bot eine FPU-Einheit, die sogar schon direkt im Prozessorkern integriert war – es gab allerdings noch eine Zeitlang Varianten ohne FPU.

# So funktionieren Pipelines und OoO

Um eine einzige Funktion auszuführen, benötigte der Urvater der heutigen Prozessoren, der Intel 8086, unzählige Takte abhängig von der Art der Instruktion. Der Fachbegriff dafür lautet CPI (Clockcycles per Instruction). Dieser Wert beschreibt die Anzahl der Takte, die benötigt werden, um eine Instruktion auszuführen.

Dem grundsätzlichen Problem der langwierigen Befehlsverarbeitung versucht man mit dem Fließbandprinzip zu begegnen; wie einst bei Henry Ford erwies es sich als zweckmäßig, die Verarbeitung einer Instruktion in mehrere, etwa gleich große Einzelschritte zu zerlegen (detaillierte Erklärung im Kasten rechts).

## Pipelining

Als erster Mikroprozessor mit einer (einstufigen) Pipeline gilt der MOS Technology 6502 von 1975. Aber auch in der x86-Welt war bereits der allererste IBM-kompatible Prozessor, der Intel 8086 im Jahre 1978, mit einer primitiven Form des Pipelinings ausgestattet. Seine Bus-Interface-Unit schickte die Abfolge an Instruktionen durch eine 6-Byte-Prefetch-Queue an die Execution-Einheit, sodass die Stufen Fetch und Execution parallel ausgeführt werden konnten.

## Pionier 486

Als Vater des echten Pipelinings auf den x86ern jedoch gilt der Intel 80486, der eine dreistufige Locate-Fetch-Execution-Pipeline besaß und im Jahre 1989 eingeführt wurde. Damals, als die Luft bei der CPU-Entwicklung noch nicht so dünn war wie heute, hatte die Einführung einer neuen Prozessor-Generation aufgrund der neuen Features, zu denen auch Pipelining gehörte, tatsächlich einen deutlichen Performance-Schub bei gleicher Taktfrequenz zur Folge. Heute können die CPU-Designer schon froh sein, wenn eine neue Generation 20 bis 30 Prozent weniger CPI benötigt als der Vorgänger. Das liegt mitunter natürlich daran, dass sich dieses Prinzip nicht beliebig erweitern lässt.

## Intels Pentium

Der Intel Pentium bekam zum ersten Mal zwei Pipelines. Damit ist theoretisch ein CPI-Wert von unter 1 möglich. Ein weiterer Meilenstein in Sachen Pipelines war der AMD Athlon, der satte drei Gleitkomma- und drei ALU-Pipelines besaß, insgesamt also sechs parallele Pipelines, die zwar nicht alle gleichzeitig bedient werden konnten, da nicht alle drei FPU-Pipelines mit allen Funktionen ausgestattet waren und der Dekoder nur drei parallele Ströme verarbeiten konnte – aber immerhin.

Zwei Jahre später brachte Intel den Pentium 4 auf den Markt, der in Sachen Pipelines ebenfalls einzigartig war, wenn auch auf eine andere Art und Weise. Mit bis zu 31 Pipelinestufen hatte die Netburst-Architektur des Pentium 4 die längste Pipeline eines x86-Prozessors. Die Absicht dahinter war klar: Je mehr Pipelinestufen, desto simpler die einzelnen Teilaufgaben, desto schneller wurden die Stages damit fertig und desto höher konnte die CPU getaktet werden.

Doch so einfach gestaltete es sich nicht und das einfache Verlängern der Pipelining bei gleichzeitig immer weiter steigendem Takt entpuppte sich als Irrweg.

## Wie eine Pipeline funktioniert

**Der ständig wiederkehrende Funktionsablauf in einem Prozessor besteht im Wesentlichen aus den Stufen IF (Instruction Fetch), ID (Instruction Decoding), EX (Execution) und WB (Write Back), wobei sich der der Ablaufplan im Bereich EX je nach Darstellung um einen MEM- oder Fetch-Operands-Zyklus erweitern lässt – schließlich müssen nicht nur die Instruktionen aus dem Speicher geladen werden, sondern auch noch die zu verarbeitenden Daten. Gehen wir also in unserem Beispiel einmal von der Reihenfolge IF, ID, EX, MEM und WB aus. Das sind fünf verschiedene Vorgänge, die ein Prozessor nacheinander ausführen muss, um einen einzigen Befehl zu verarbeiten – grundsätzlich spiegeln diese auch die fünf Phasen der Befehlsverarbeitung eines Von-Neumann-Rechners wider.**

### Leerlauf

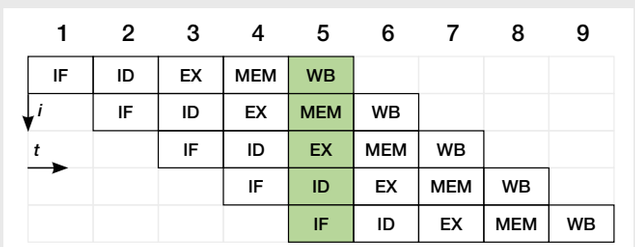
**Ein solcher vereinfachter Prozessor hätte einen CPI-Wert von fünf. Für jeden Befehl benötigt die CPU also selbst im Idealfall fünf ganze Takte, wobei dieser Idealfall höchst selten eintritt. Schließlich dauert bereits ein Zugriff auf den im Verhältnis deutlich langsameren Arbeitsspeicher etliche Dutzend bis hundert Prozessortakte. In dieser Zeit ist unser Prozessor zum Warten verdammt; eine Zeit, in der die CPU nichts weiter tun kann, als Däumchen zu drehen, bis die Infrastruktur die Daten aus dem RAM gelesen hat.**

### Fließbandprinzip

**Statt alle Einheiten an einer einzelnen Operation arbeiten zu lassen, unterteilen aktuelle Prozessoren die Aufgaben in möglichst kleine Teilbereiche – genau wie einst Henry Ford bei der Produktion von Autos. Der Trick des Pipelinings ist es nun, die Schritte IF, ID, EX, MEM und WB genau wie am Fließband zwar nach wie vor hintereinander auszuführen, jedoch den nächsten IF-Arbeitsschritt bereits dann zu beginnen, wenn Stufe 2 der Pipeline noch mit ID beschäftigt ist. So wird die Pipeline bei jedem Takt mit einer neuen Instruktion gefüttert.**

### Effizientere Verarbeitung

**Zwar dauert es genau wie am Fließband nach wie vor genauso viele Taktzyklen wie ohne Pipeline, bis die erste Instruktion abgearbeitet ist. Anschließend jedoch wird der Prozessor nach jedem Takt wieder mit einer Instruktion fertig, während der Non-Pipelined-Prozessor hier erst wieder mit seiner nächsten Instruktion beginnen würde. Durch die CPU-Pipeline haben wir es also geschafft, den CPI-Wert unseres Beispielprozessors von bisher bestenfalls 5 auf bis zu 1 zu verringern.**



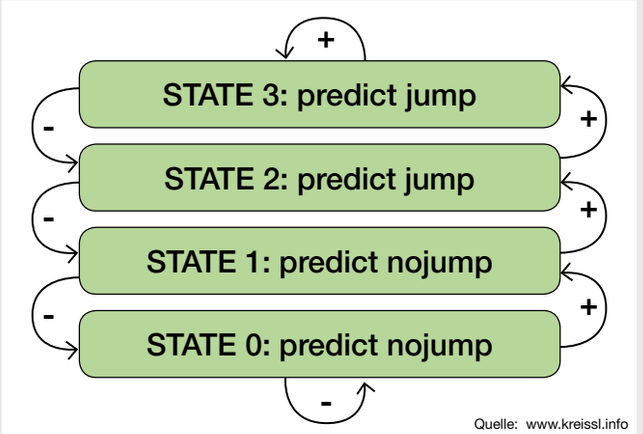

**Die fünfstufige Pipeline unserer Beispiel-CPU: Ab dem fünften Takt wird die CPU mit jedem Zyklus mit einer Instruktion fertig.**

Bild: Intel

Intels Pentium (1) brachte erstmals zwei Pipelines; eine davon konnte aber nur einfache Aufgaben berechnen (V-Pipe), während die andere universell einsetzbar war (U-Pipe).

## Spekulative Ausführung

Speculative Execution: Wird ein Sprung zweimal hintereinander ausgeführt, ändert der Predictor seine Vorhersage von Nojump auf Jump.




Quelle: www.kreissl.info

Beim K6 kombinierte AMD den Befehlssatz der fünften CPU-Generation mit der Out-of-Order-Fähigkeit der sechsten Generation.

### Pipeline-Probleme

Ein Prozessor ist kein streng sequenzieller Automat, bei dem alle Vorgänge vorhersehbar sind. Ein Programm ist vor allem eine Ansammlung von unzähligen Entscheidungsfindungen. Die Wenn-dann-Struktur eines Programms steht jedoch in direkter Konkurrenz zu einer möglichst tiefen Pipeline. Das ist immer dann kritisch, wenn beispielsweise die dritte Instruktion vom Ergebnis der ersten Instruktion abhängt. Bereits in unserer äußerst kurzen Beispiel-CPU mit fünf Pipeline-Stufen ist das ein Problem, denn wenn die Instruktion Nr. 3 in die ID-Stufe geladen wird, ist die Instruktion Nr. 1 noch nicht fertig berechnet.

Es gibt nun zwei Möglichkeiten, eine CPU in so einem Fall verfahren zu lassen: Entweder wartet die CPU mit dem Start der dritten Instruktion so lange, bis das Ergebnis der ersten Instruktion feststeht. Das ist die einfachste Art der Implementierung, aber auch die langsamste. Im schlimmsten Fall arbeitet der Prozessor wie ein Prozessor ohne Pipeline, weil er ständig auf das Ergebnis einer vorherigen Stufe wartet.

### Spekulative Ausführung

Die andere Möglichkeit der Implementierung lautet, die Pipeline einfach auf Verdacht arbeiten zu lassen. Wenn die Instruktion 1 zwei Takte später das Ergebnis hat, das Instruktion 3 „angenommen“ hat, bleibt die heile Welt des Pipelinings erhalten.

Übel wird es allerdings, wenn sich die CPU „verschätzt“ hat und die Instruktion 3 unter falschen Voraussetzungen auf die Reise geschickt hat. Dann muss die komplette Pipeline geleert („Flush“) werden, alle in der Pipeline befindlichen Stationen müssen ihre Arbeit auf den Müll werfen. Wenn das permanent geschieht, kann eine Pipelined-CPU sogar langsamer arbeiten als eine Non-Pipelined-CPU, da das Flushen einer Pipeline selbst bereits etliche Takte für sich in Anspruch nimmt.

### Branch-Prediction

Trotzdem arbeiten fast alle Pipelined-CPUs mit diesen Berechnungen auf Verdacht. Allerdings haben die CPU-Entwickler ihren Babys ein Feature mit auf den Weg gegeben, das dafür sorgt, dass ein Pipeline-Flush so selten wie möglich nötig ist. Dieses Feature nennt sich Branch-Prediction-Unit, zu Deutsch: Sprungvorhersage. Eine Sprungvorhersage (eigentlich ist dies eine falsche Übersetzung, „branch prediction“ lautet korrekt übersetzt „Abzweigungsvorhersage“ oder „Verzweigungsvorhersage“) versucht zu erraten, welche Richtung der Programmfluss bei einem bedingten Sprung einlegen wird. Hierbei werden die verschiedenen Verzweigungsmöglichkeiten des Programmflusses genauestens geprüft und nach Wahrscheinlichkeiten gewichtet. Die Verzweigung mit der höchsten Wahrscheinlichkeit wird anschließend als weiterer Verlauf des Programms definiert und die Pipeline mit den entsprechenden Instruktionen gefüllt.

Doch wie das Wort „Wahrscheinlichkeit“ schon zeigt, gibt es keine Garantie, dass der Programmfluss die vorhergesagte Richtung auch wirklich einschlägt. Somit wird es immer den Fall geben, dass aufgrund eines Versagens der Sprungvorhersage-Logik die Pipeline ausgespült und neu gefüllt werden muss. Die perfekte Sprungvorhersage mit 99 Prozent Treffsicherheit (oder gar mehr) hat bisher noch niemand konstruieren können, Informatiker und Ingenieure rund um den Globus beschäftigen sich mit dem Modell der „Sprungvorhersage“.

## Branch Target Buffer

Besser als eine bloße Sprungvorhersage ist eine Sprungzielvorhersage. Sobald man in der ID-Stage erkennt, dass es sich um einen Sprung handelt, kann man prüfen, ob dieser Sprung schon einmal stattfand, und gegebenenfalls sein Sprungziel aus einem Puffer holen. Somit kann man den Programmzähler sofort auf dieses Sprungziel stellen und die dortigen Instruktionen in die Pipeline laden. Dies geschieht in der Regel mithilfe eines Branch Target Buffers (BTB) oder eines Branch Target Adress Cache. Diese Tabelle enthält Vorhersageinformationen, Zieladressen und Tags.

## Superskalarität

Mit einem strikt nach von Neumann arbeitenden Prozessor würde man niemals eine IPC > 1 erreichen können (bzw. CPI < 1). Die Tatsache, dass heutige Prozessoren eine IPC von deutlich mehr als 1 haben, beruht auf der Tatsache, dass die Entwickler die von-Neumann-Architektur nach eigenen Vorstellung etwas verbogen haben und den Prozessoren, beginnend mit dem Intel Pentium (586), die superskalare Arbeitsweise eingeimpft haben.

Dabei geht es nicht ausschließlich um das Pipelining. Eine superskalare CPU unterscheidet sich von einem nicht-superskalaren Prozessor hauptsächlich im Leitwerk. So teilt das Leitwerk einer Recheneinheit nicht mehr lediglich eine Instruktion pro Takt zu, sondern gleich mehrere unterschiedliche Instruktionen für mehrere unterschiedliche Recheneinheiten. Es gibt damit im Grunde nicht nur eine Pipeline, sondern mehrere parallele. Der Prozessor ist damit in der Lage, mehrere Instruktionen parallel abzuarbeiten, solange diese oder ihre Ergebnisse nicht voneinander abhängig sind. Somit kann eine zweifach superskalare CPU bei gleichem Takt im Idealfall doppelt so viele Instruktionen pro Sekunde abarbeiten wie ein serieller Prozessor. Um dies zu erreichen, muss beispielsweise der Dekoder mehr als nur eine Instruktion pro Taktzyklus dekodieren können. Es müssen mehrere voneinander unabhängige Recheneinheiten vorhanden sein. Die erste Implementierung superskalarer Arbeitsweise fand in der fünften x86-Generation statt, im Intel 586 alias Pentium.

## Out-of-order-Execution

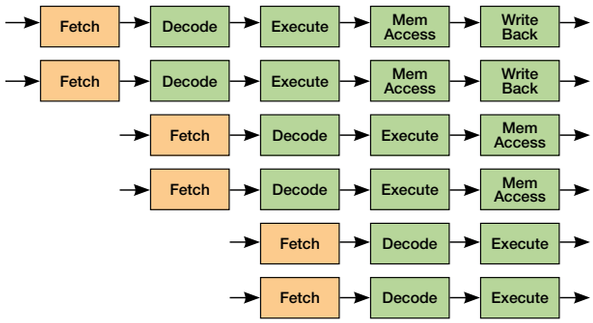
Mit der Superskalarität wurde es noch wichtiger, die Recheneinheiten optimal auszulasten. Zu diesem Zweck ersannen die Entwickler die „Out-of-Order Execution“ (kurz: OoO, zu Deutsch: „Ausführung außerhalb der Reihenfolge“). Sie bietet einen enormen Zuwachs an Geschwindigkeit, denn so können Instruktionen bereits ausgeführt werden, wenn sie noch gar nicht an der Reihe sind.

Ein Beispiel: Nehmen wir an, dass eine Instruktion Daten aus dem Speicher einlesen muss. Das führt dazu, dass der Prozessor für mehrere hundert bis tausend Taktzyklen auf die Daten warten muss. Gibt es jedoch in der Pipeline Instruktionen, die von der derzeit auf Daten aus dem Speicher wartenden Instruktion unbabhängig sind, kann der Prozessor die nötige Wartezeit mit der Ausführung dieser Instruktionen überbrücken.

Out-of-Order Execution wurde in der x86-Welt erstmals mit dem Pentium Pro eingeführt. Dennoch gibt es auch heute noch CPUs, die zugunsten einer niedrigen Leistungsaufnahme auf OoO verzichten – Intels erste Atom-CPU ist ein Beispiel dafür. Ein OoO-Prozessor braucht zudem immer eine Register-Renaming-Einheit, mit der die originalen Register „simuliert“ werden.

## Wie ein superskalarer Prozessor arbeitet

**Eine fünfstufige Pipeline mit zweifacher Superskalarität. Hier können zwei Befehle gleichzeitig ausgeführt werden.**




## Pipelining und Out-of-Order-Execution

**Im unserem abgebildeten Beispiel laufen drei Befehle parallel; erschwerend kommt hinzu, dass die grüne Instruktion (addiere F1 und F3 und schreibe das Ergebnis in F4) vom Ergebnis der roten abhängig ist. Im oberen Teil der Grafik muss die grüne Instruktion warten, bis das Ergebnis der roten vorliegt. Es müssen daher zwei Wartezyklen (S, stall) eingelegt werden. Dies blockiert auch die blaue Operation. Dennoch sind alle drei Aufgaben bereits nach 9 Taktzyklen erledigt – ohne Pipelining hätten die drei Instruktionen 15 Taktzyklen benötigt.**

**In der unteren Hälfte ist die gleiche Aufgabe auf einem Out-of-Order-Execution-fähigen Chip zu sehen. Hier wurde die ursprünglich dritte Instruktion (blau) vorgezogen. Sie kann nun ohne Wartezeit bearbeitet werden. Als Nebeneffekt ergibt sich, dass die grüne Instruktion nun nur noch einen Takt auf das Ergebnis der roten Instruktion warten muss.**

| Taktzyklus | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| IF | ID | EX | M | WB | | | | |
| | IF | ID | S | S | EX | M | WB | |
| | | IF | S | S | ID | EX | M | WB |

add F1, F2 => F3
add F1, F3 => F4
sub R1, R2 => R3

| Taktzyklus | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| IF | ID | EX | M | WB | | | | |
| | IF | ID | EX | M | WB | | | |
| | | IF | ID | S | EX | M | WB | |

**Legende:**
**IF** (Instruction Fetch): Der nächste Befehl wird geholt. **ID** (Instruction Decode): Der Befehl wird dekodiert. **EX** (Execute): Der Befehl wird ausgeführt. **M** (Memory): Der Speicher wird adressiert. **WB** (Write Back): Das Ergebnis der Berechnung wird geschrieben. **S** (Stall): Die Pipeline läuft leer.

# So funktionieren Caches

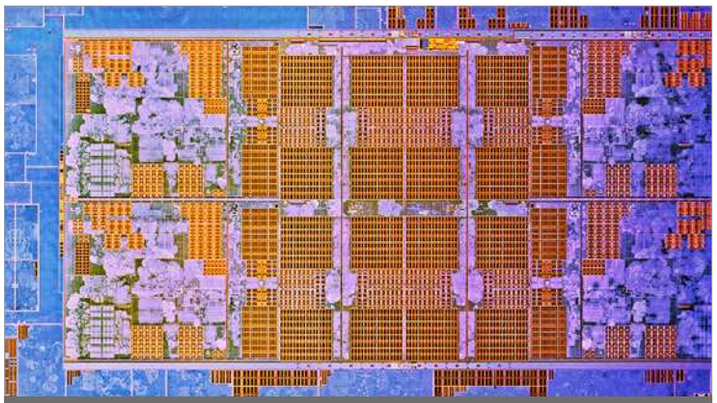


Bild: AMD

Cache kann man aufgrund der uniformen Struktur leicht auf dem Wafer oder entsprechenden Bildern („Die-Shots“) erkennen (hier: Ryzen).

Bild: Intel

Mit dem Pentium III Coppermine und dem AMD K6-III konnte der L2-Cache im Chip untergebracht werden – dort ist er bis heute zentraler Bestandteil jeder CPU

Der klassische von-Neumann-Rechner besitzt keinen Cachespeicher, ja nicht einmal Register. Sämtliche Operationen werden direkt aus dem Speicher heraus ausgeführt – und das dauert!

## Langsamer Speicher

Verantwortlich dafür ist zum einen die Diskrepanz zwischen CPU- und Speichertakt. Während aktuelle Prozessoren mit 4 GHz Taktfrequenz und mehr arbeiten, bleibt selbst die Bandbreite schnellster DDR4-Speicher deutlich unter dem, was die CPU verarbeiten könnte.

Damit könnte man noch leben. Doch noch schlimmer sind die miserablen Zugriffszeiten („Latenzen“) von DRAM-Speicher: Von der Anforderung von Daten bis zur Lieferung vergehen je nach Plattform zwischen 150 und 500 Prozessortakte! Im schlimmsten Fall (wenn OoO oder andere Funktionen nicht greifen), kann der Prozessor in dieser Zeit nichts anderes tun als warten.

## Cache-Geschichte

Dieser Missstand wurde bereits früh erkannt. Schon 8088-Systeme besaßen Cache-Bausteine aus SRAM-Speicher auf den Mainboards, um den Zugriff auf häufig verwendete Codestellen und Daten zu beschleunigen.

Der Trick dabei: Daten, die bereits einmal aus dem Speicher geholt wurden, werden in einen Puffer geschrieben. Benötigt die CPU dieselben Teile wenig später noch einmal, was relativ häufig vorkommt, braucht sie nicht erneut darauf zu warten. Die Daten müssen dann nicht mehr aus dem RAM gelesen, sondern können relativ schnell aus dem Pufferspeicher bereitgestellt werden.

## Cache-Hierarchie

Allerdings sind klassische Cache-Bausteine auf den Mainboards immer noch relativ langsam, arbeiten sie doch meist mit der Taktfrequenz der Infrastruktur, während die CPU meist 10 bis 20 Mal so hoch taktet. Daher integrierte Intel beim 80486 zum ersten Mal zusätzlichen Cache auch direkt in die CPU. Der gestufte Cache war erfunden. Beim 486er waren es meist 2x 4 KiByte Puffer in der CPU als Level-1-Cache sowie größerer (damals meist 256 kiByte), aber langsamerer Level-2-Cache auf dem Mainboard.

Der Vorteil des integrierten Caches: Die Wege sind kürzer, die Latenzzeiten sinken gewaltig, der Cache arbeitet mit vollem CPU-Takt und die Anbindung kann breiter ausgeführt werden, als es die Infrastruktur vorgibt. Mit stetig verbesserten Produktionstechniken und verkleinerten Strukturen konnte mit dem Pentium III Coppermine und AMD K6-III später auch der L2-Cache direkt in die CPU integriert werden. Seit dieser Zeit besitzt praktisch jede CPU einen On-Die-Level-1- und Level-2-Cache.

## L3- und L4-Cache

Später kam dann noch eine weitere Hierarchie hinzu: der L3-Cache. Einer der ersten bekannteren Desktop-Prozessoren war der Pentium 4 Extreme Edition mit Gallatin-Kern, der im Jahr 2003 erschien; dieser zielte auf leistungshungrige Anwender, vor allem Spieler. Seinen 512 KiByte L2- standen zusätzlich 2 MiByte L3-Cache zur Seite. Heute sind L3-Cachegrößen von 8 MiByte und mehr üblich.

Eine Zeitlang, genauer gesagt von zirka 2013 bis 2017, gab es auch Prozessoren mit L4-Cache. Intels Broadwell-, Haswell- und Skylake-Architektur sahen hierfür allerdings kein SRAM (dazu gleich mehr), sondern eDRAM (Embedded-DRAM) vor, das auf den Chip gepackt und sowohl von der Onchip-Grafikeinheit als auch von der CPU genutzt werden konnte. Zwar profitierten einige Programme wie 7-Zip oder Spiele wie Starcraft 2 enorm von dem On-Package-Speicher; langfristig durchsetzen konnte er sich bis heute aber nicht – L4-Cache verschwand gegen 2017 mit den letzten eDRAM-Prozessoren.

## SRAM-Speicher

Implementiert wird Cache durch sogenanntes SRAM. Statisches RAM (engl. Static Random Access Memory) bezeichnet einen Typ von flüchtigen Speicherbausteinen. Im Gegensatz zu DRAMs müssen außer der Betriebsspannung keine Signale zum Auffrischen erzeugt werden, um die Daten zu erhalten. Eine SRAM-Speicherzelle besteht aus zwei Transistoren, die als Flipflop geschaltet sind, und weiteren Transistoren für die Steuerung des Schreib- und Lesevorgangs. In der Regel sind für eine Speicherzelle sechs CMOS-Transistoren nötig.

Jede Speicherzelle bildet 1 Bit ab. SRAM hat sehr geringe Zugriffszeiten und ist damit als Cachespeicher prädestiniert. Der Nachteil im Vergleich zu DRAMs ist der höhere Flächenbedarf auf dem Wafer bei gleicher Speicherkapazität und der damit vielfach höhere Preis.

## Cache-Arbeitsweisen

Je nach Prozessor ist ein zweigeteilter Split-Cache oder ein einheitlicher Unified-Cache verbaut, der Daten und Befehle gemischt speichert. Die aktuellen CPUs von AMD und Intel verwenden Split-Caches für den L1- und Unified-Cache für den L2-/L3-Zwischenspeicher. Aus Platzgründen kann der L1-Cache in der CPU nicht sehr groß sein. Je nach Philosophie des Prozessorherstellers bewegt er sich zwischen 32 und 128 kiByte pro Kern.

## Cache-Organisation

Der Cache wird in Zeilen (Cache Lines) oder Sets eingeteilt. Erfolgt ein Lesezugriff auf einen Bereich im Arbeitsspeicher, dann füllt die CPU eine Cache-Zeile mit dem Inhalt aus diesem Speicherblock. Es gibt verschiedene Organisationen von Caches. Der einfachste ist der Direct-mapped-Cache. Hier ist jedem Set nur ein Cache-Eintrag zugeordnet. Der Nachteil des Direct-mapped-Caches ist, dass nur Speicherbereiche in der Größe des Caches gepuffert werden können. Wird auf Speicherbereiche außerhalb dieses Bereiches zugegriffen, ist der gesamte Inhalt des Eintrages wertlos und muss neu geladen werden.

## Assoziativer bzw. Mehrweg-Cache

Hier eilen teil- oder mehrwegassoziative Caches zu Hilfe. Die Adressen werden dabei in mehreren Ebenen gespeichert und miteinander verknüpft. Erfolgt ein Zugriff auf eine Adresse außerhalb des aktuellen Blocks, die aber dieselbe Anfangsadresse enthält, so wird der Speicherblock nicht komplett ausgetauscht, sondern in einer anderen Ebene gespeichert.

Ein gern verwendeter Vergleich ist der Briefträger: Er sortiert die Post nicht nur in Straßenzüge (Tags), sondern obendrein auch nach der Straßenseite. Bei einem Abstecher in die Seitenstraße greift er sich nur das Briefbündel für eine Straßenseite und legt das zweite Bündel der gegenüberliegenden Straßenseite der alten Straße weg. Wenn er

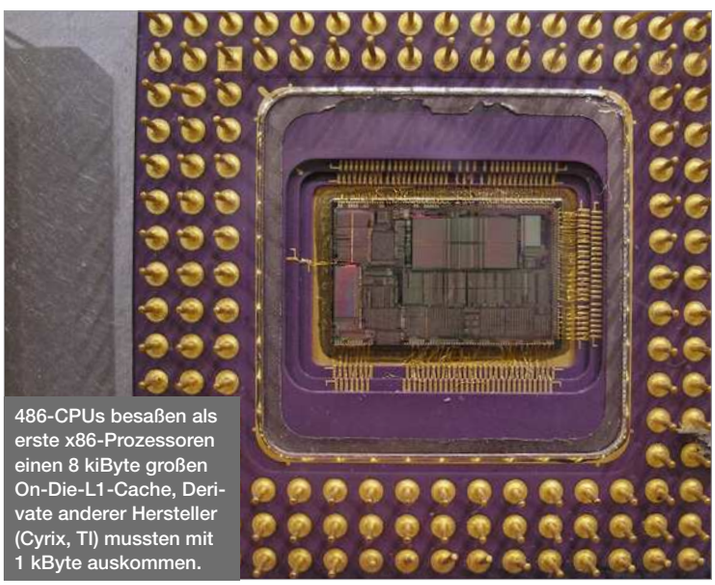
# Cache-Geschwindigkeiten

**Die Geschwindigkeit des Zwischenspeichers nimmt immer mehr ab, je weiter dieser vom CPU-Kern entfernt ist – und das ist durchaus wörtlich zu nehmen! Bei einem Takt von 4 GHz können Elektronen aufgrund der maximalen Driftgeschwindikeit nur noch ca. 2,5 Millimeter pro Takt zurücklege (dies ist zudem ein rein theoretischer Wert ohne alle anderen praxisrelevanten Faktoren)**

**Wichtig auch: Die Transfergeschwindigkeit sagt nichts über die Latenzen aus; im Einzelfall kann der Transfer deutlich langsamer, aber auch schneller erfolgen als im Benchmark-Programm gemessen.**

### Cache- und Speichertransfer bei 3,7 GHz CPU-Takt

| L1-Cache | Lesen | Schreiben | Kopieren |
|---|---|---|---|
| Skylake | 918 GB/s | 462 GB/s | 921 GB/s |
| Broadwell | 903 GB/s | 461 GB/s | 918 GB/s |
| Haswell | 920 GB/s | 461 GB/s | 918 GB/s |
| **L2-Cache** | **Lesen** | **Schreiben** | **Kopieren** |
| Skylake | 388 GB/s | 251 GB/s | 354 GB/s |
| Broadwell | 359 GB/s | 148 GB/s | 215 GB/s |
| Haswell | 360 GB/s | 148 GB/s | 218 GB/s |
| **L3-Cache** | **Lesen** | **Schreiben** | **Kopieren** |
| Skylake | 250 GB/s | 160 GB/s | 232 GB/s |
| Broadwell | 188 GB/s | 123 GB/s | 141 GB/s |
| Haswell | 195 GB/s | 140 GB/s | 156 GB/s |
| **L4-Cache (EDRAM)** | **Lesen** | **Schreiben** | **Kopieren** |
| Skylake | - | - | - |
| Broadwell | 51,2 GB/s | 36,4 GB/s | 43,2 GB/s |
| Haswell | - | - | - |
| **Arbeitsspeicher** | **Lesen** | **Schreiben** | **Kopieren** |
| Skylake | 32,0 GB/s | 32,8 GB/s | 31,8 GB/s |
| Broadwell | 27,7 GB/s | 25,0 GB/s | 28,7 GB/s |
| Haswell | 24,3 GB/s | 25,1 GB/s | 23,6 GB/s |

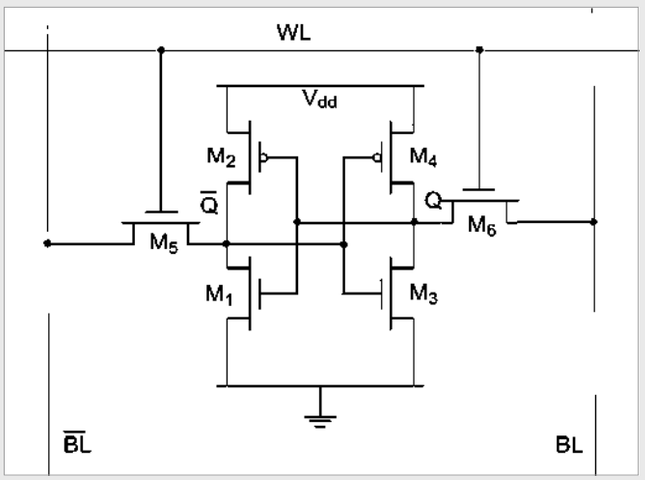


486-CPUs besaßen als erste x86-Prozessoren einen 8 kiByte großen On-Die-L1-Cache, Derivate anderer Hersteller (Cyrix, TI) mussten mit 1 kByte auskommen.

Bild: DEC

# Wie Cache-Speicher intern aufgebaut ist

**Ein SRAM-Cache benötigt zur Speicherung eines einzigen Bits satte sechs Transistoren. Vier Transistoren speichern die Information, die restlichen werden zum Ein- und Auslesen benötigt. Daneben gibt es aber auch eine Reihe alternativer und verbesserter SRAM-Designs; in der Nehalem-Architektur beispielsweise kommen SRAM-Zellen mit acht Transistoren zum Einsatz, die für einen geringeren Energiebedarf sorgen sollen.**




# Warum es für kurze Zeit Slot-Prozessoren gab

**Der erste AMD Athlon besaß ebenso wie einige Pentium II und III einen L2-Cache in Form von auf der CPU-Platine aufgelöteten SRAM-Chips. Dazu war damals die Slot-Bauweise nötig. Später passte der Cache aufgrund des besseren Herstellungsprozesses auf den Chip – die Slot-Gehäuse verschwanden wieder.**

Bild: AMD

aus der Seitenstraße zurückkommt, hat er immer noch das Bündel der Seite in der Hand, bei der er abgebogen ist.

Level-1-Caches aktueller CPUs sind in der Regel 2-Wege-assoziativ, Level-2-Caches meist 2- bis 8-Wege-assoziativ und Level-3-Caches 16-Wege-assoziativ. Konkret hat beispielsweise Intels Skylake 4-Wege-assoziative L2-Caches (256 KiByte), während AMD bei Ryzen auf 8-Wege-Assoziativität setzt (512 KiByte). Diese Parameter sind aber immer wieder Gegenstand von Optimierungsversuchen der CPU-Architekten.

## Instruktions-Cache

Intels Pentium 4 brachte einen Sonderfall des L1-Instruction-Cache, den sogenannten „Trace Cache". Seit dem 6x86 dekodieren x86-Prozessoren jede Instruktion vor ihrer Ausführung in RISC-artige Mikro-Operationen. Dieser Vorgang der Dekodierung kostet natürlich Rechenzeit sowie ein wenig Energie.

Um diese zu sparen, speicherte der Pentium 4 zirka 12.000 RISC-artige, bereits dekodierte Mikro-Operationen (µOps) im L1-Trace-Cache. Der Trace Cache verschwand wenig später wieder, erlebte aber mit der Sandy-Bridge-Architektur ein Comeback und ist heute Bestandteil fast aller Intel– und AMD-Prozessoren. Ein „Trace" oder „Dekodierter-Befehls"-Cache exisitiert übrigens fast immer zusätzlich zum L1-Instruktionscache.

## Cacheverwaltung

Zusätzlich zu den erwähnten Varianten unterscheidet man zwischen inklusiver und exklusiver Cacheverwaltung. Bei der inklusiven Verwaltung nehmen die einzelnen Cache-Stufen keine Rücksicht darauf, ob Daten bereits in der anderen Stufe liegen oder nicht. Es kann also vorkommen, dass Daten sowohl im L1- als auch im L2-Cache liegen (oder analog dazu im L2- und L3-Cache) und damit mehrfach vorhanden sind. Angesichts der geringen Größe der Caches eigentlich eine böse Verschwendung. Allerdings arbeitet ein inklusiver Cache aufgrund des niedrigeren Verwaltungsaufwandes in der Regel mit sehr niedrigen Latenzzeiten.

Beim exklusiven Cache dagegen können Daten niemals gleichzeitig in L1 und L2 liegen und damit Platz verschwenden. Die beiden Cachestufen werden so verwaltet, als wären sie ein großer Cache-Pool – mit dem einzigen Unterschied, dass der L1 noch einmal schneller arbeitet als der L2. Der Nachteil dabei: Die Latenzen sind etwas höher. Bei aktuellen CPU-Architekturen der x86-Welt sind die L3-Caches meist inklusiv ausgelegt.

## Cachegröße

Je größer der Cachespeicher, desto mehr Daten kann die CPU im Puffer halten und desto größer ist die Wahrscheinlichkeit, dass sich ein benötigtes Datenpaket bereits im Cache befindet. Aktuell scheint der beste Kompromiss zwischen Cache-Größe und Platzbedarf auf dem CPU-Kern bei zirka sechs bis acht MiByte für den L3-Cache zu liegen; fast alle Mittelklasse-Prozessoren, egal ob bei AMD oder Intel, bewegen sich in diesem Rahmen.

Bei den L2-Caches pendeln die Größen zwischen 64 und 512 kiByte pro Kern. Der L1-Cache ist noch kleiner und meist 16 bis 64 kiByte groß. Die Entwickler experimentieren aber immer wieder mit den Größen und münzen ein nicht ganz ausgeschöpftes Transistorbudget gerne mal in zusätzlichen Cache um. AMDs Ryzen beispielsweise hat nun wieder mehr Datencache (32 statt 16 KiByte).

# So funktionieren x86-Erweiterungen

Ob FMA, AVX oder SGX; all diese Begriffe, die aus dem Wortschatz eines PC-Enthusiasten nicht mehr wegzudenken sind, haben eines gemeinsam: Es sind Versuche der Prozessorhersteller, die Unzulänglichkeiten der ursprünglichen x86-Architektur auszugleichen.

## Architektur-Defizite

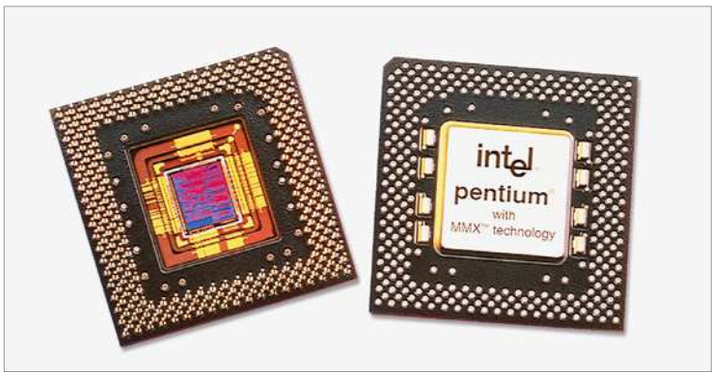
Im Gegensatz etwa zu den CPUs der Amiga-Rechner in den 80er-Jahren waren der Intel 8086 und seine Nachfolger nie dafür gedacht, 3D-Spiele auf den Bildschirm zu zaubern oder wie die Apple- bzw. Alpha-Prozessoren komplexe Grafikberechnungen in Echtzeit durchzuführen und mit Vektoren im dreidimensionalen Raum zu hantieren.

Dass es im Endeffekt anders gekommen ist, verdankt der PC vor allem seiner weiten Verbreitung, findigen Programmierern, die es verstanden haben, mit intelligenter Programmierung beispielsweise auch Rennsimulationen 286-tauglich zu machen, sowie etlichen Hardware-Entwicklungen in diese Richtung (3D-Add-on-Karten, MPEG-Karten, etc).

Aufgrund seiner nur bedingt tauglichen Konzeption suchten die Ingenieure bei Intel und AMD aber schon früh nach einer Möglichkeit, dem PC-Prozessor weitere Features mit auf den Weg zu geben, ohne zu bestehender Hard- und Software inkompatibel zu werden.

## MMX

Den Anfang machte Intel 1996 mit dem Pentium MMX. Die MMX-Unit war die erste SIMD-Einheit in einem x86-Prozessor. Dieses Rechenwerk kann mit einer Instruktion mehrere Daten miteinander verarbeiten (SIMD = Single Instruction, Multiple Data). Dazu spendierte man der CPU acht zusätzliche Register namens MM0 bis MM7, welche den neuen Datentyp „Packed“ aufnehmen konnten. Diese Register konnte man nun mit verschiedenen Packed-Varianten füllen. Zwei 32-Bit-Zahlen, vier 16-Bit-Zahlen oder acht 8-Bit-Zahlen. Addiert man nun beispielsweise MM0 mit MM1 (oder führt eine beliebige Rechenoperation aus), so hat man mit einer Instruktion im Idealfall acht Zahlen miteinander verrechnet statt nur eine mit dem herkömmlichen Design.

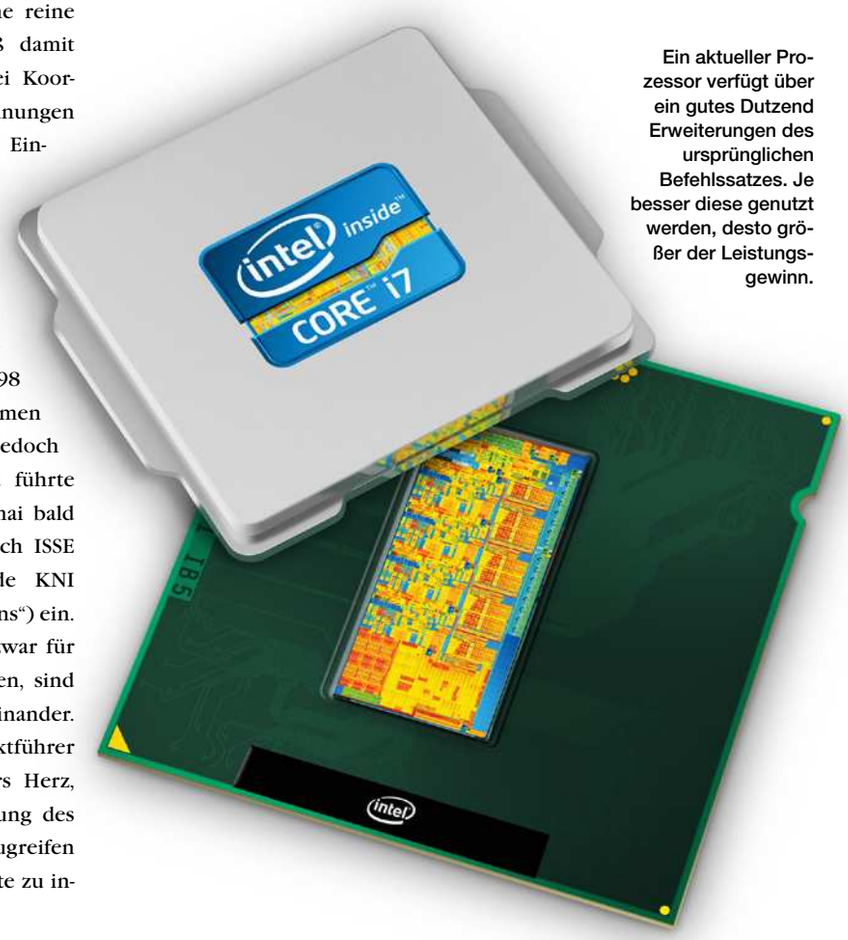



Die erste Befehlssatz-Erweiterung in der x86-Welt brachte Intels Pentium MMX – bis heute weiß niemand, wofür MMX eigentlich steht, für „Multimedia Extensions“ jedenfalls nicht, wie Intel stets bekräftigte.

## 3DNow und SSE

Bald jedoch stellte sich heraus, dass SIMD zwar eine gute Idee war, MMX im Speziellen sie aber am Thema vorbei in die Praxis umsetzte. Die MMX-Unit war eine reine Ganzzahl-Einheit und ließ damit die nötige Genauigkeit bei Koordinaten- und Grafikberechnungen vermissen. Es musste eine Einheit her, die auch Gleitkomma-Berechnungen anstellen konnte.

Die erste Gleitkomma-SIMD-Einheit präsentierte AMD im K6-2 1998 unter dem Markennamen 3DNow. Intel ließ sich jedoch nicht zweimal bitten und führte mit dem Pentium III Katmai bald darauf SSE (zu Anfang noch ISSE genannt; Entwicklungscode KNI für „Katmai New Instructions“) ein. 3DNow und SSE wurden zwar für denselben Zweck entworfen, sind jedoch inkompatibel zueinander. Intel brachte es als Marktführer damals (noch) nicht übers Herz, eine bestehende Entwicklung des Hauptkonkurrenten aufzugreifen und in die eigenen Produkte zu in-



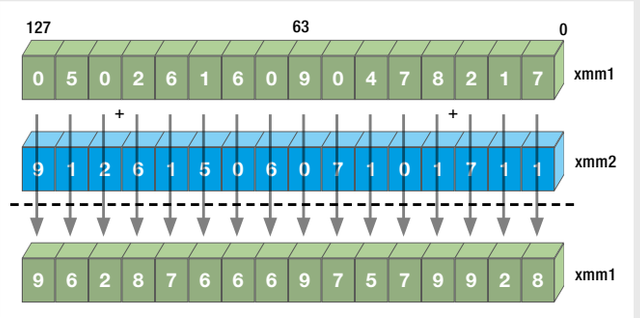

Ein aktueller Prozessor verfügt über ein gutes Dutzend Erweiterungen des ursprünglichen Befehlssatzes. Je besser diese genutzt werden, desto größer der Leistungsgewinn.

# Wie SSE-Register aufgebaut sind

**SSE erhöht die Rechenleistung drastisch: Die zusätzlichen Register sind „breiter“, was bedeutet, dass sie mehr Bits aufnehmen und verarbeiten können. So können entweder sehr große/lange Zahlen (beispielsweise mit vielen Nachkommastellen) oder mehrere kurze Zahlen in einem Schritt verrechnet werden.**

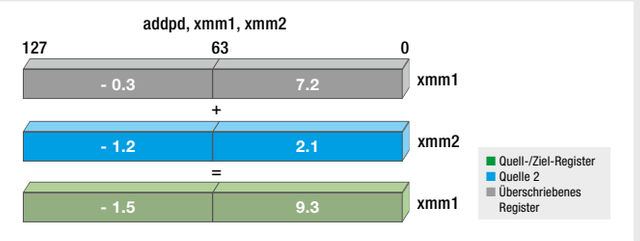
## SSE-Register

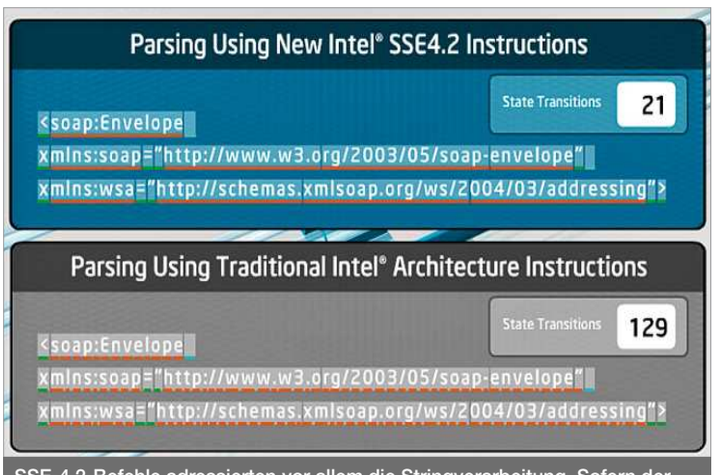
**Mit SSE ist es möglich, bis zu 16 8-Bit-Zahlen in einem Aufwasch miteinander zu verrechnen.**




## SSE-2-Register

**Entscheidende Innovation von SSE 2 war die Verarbeitung von 64-Bit-Double-Precision-Floatzahlen.**




SSE-4.2-Befehle adressierten vor allem die Stringverarbeitung. Sofern der Programmierer bzw. Compiler sie nutzt, beschleunigen diese das Parsing von zum Beispiel XML-Dateien drastisch.

tegrieren. So konkurrierten 3DNow und SSE jahrelang miteinander, wobei sich SSE schließlich durchsetzte. SSE hat den Vorteil, auf einen Satz von 128 Bit breiten Datenregistern zurückgreifen zu können, während 3DNow auf 64-Bit-Größe beschränkt war. Damit konnte man bei SSE zum Beispiel vier Single-Precision-Floatzahlen gleichzeitig verrechnen, während mit 3DNow nur zwei möglich sind. Das war das Todesurteil für 3DNow und heute ist die Erweiterung auch kein Bestandteil mehr von AMD-Prozessoren.

Allen SIMD-Befehlen ist gemein, dass sie explizit vom Programmierer genutzt werden müssen. Im Zweifelsfall muss der Programmierer den kompletten Code in Assembler schreiben, eine Vorgehensweise, die viel Mühe, Zeit und damit Geld kostet.

## Erweiterung SSE 2

Mit SSE kam die Entwicklung von Befehlssatz-Erweiterungen erst richtig in Schwung. Der Pentium 4 brachte schon SSE 2. SSE 2 ermöglichte es nun auch, Doule-Precision-Floatzahlen, also Gleitkommazahlen mit 64-Bit-Genauigkeit, zu berechnen, während SSE und 3DNow hier auf 32-Bit-Zahlen beschränkt waren.

Bei der Verdoppelung der Datenregister-Breite blieb es jedoch nicht. Zusätzlich spendierte Intel 144 neue Befehle, was die Mächtigkeit und die Bandbreite an möglichen Einsatzgebieten deutlich erhöhte. Gerade was die Berechnung von Videomaterial betrifft, bietet SSE 2 dem Programmierer einige Werkzeuge „in Hardware“ an, die entweder das Encoding beschleunigen können oder die Wiedergabe mit niedrigerer CPU-Last möglich machen.

## Die Nachfolger: SSE 3, SSSE 3, SSE 4

Deutlich weniger spektakulär war die Einführung des SSE-3-Befehlssatzes, einer erneuten Erweiterung der SSE-Basis. SSE 3 erweiterte den SSE 2-Befehlssatz um zusätzliche 13 Instruktionen zur Wandlung von Gleitkommazahlen in Ganzzahlen, für komplexe Arithmetik, zur Video-Kodierung, zur Unterstützung der Grafik-Aufbereitung und zur Thread-Kommunikation bei den Intel-Hyper-Threading-Prozessoren. Erstmals zum Einsatz kam SSE 3 beim Intel Pentium 4 Prescott (PNI: Prescott New Instructions). SSSE 3 (Supplemental Streaming SIMD Extensions 3) bezeichnet die mit Intels Core-Architektur eingeführten Erweiterungen des SSE-3-Befehlssatzes.

Während SSSE 3 mit 32 neuen Befehlen nur eine geringfügige Erweiterung zu SSE 3 darstellt, ist SSE 4 mit 50 neuen Opcodes eine substanzielle Verbesserung. SSE 4 wurde mit den 45-nm-Prozessoren beginnend ab dem Penryn eingesetzt. SSE 4 besteht aus zwei „Subsets“, nämlich SSE 4.1 (ab Penryn) und SSE 4.2 (ab Nehalem). SSE 4a dagegen ist eine AMD-spezifische Erweiterung, die ab dem Barcelona-Kern zum Einsatz kommt.

## SSE 5 und AES

Mit SSE 4 war es erst einmal vorbei mit den „Streaming Extensions“ – SSE 5 gabe es nie, auch wenn dies vorschnell angekündigt wurde. AMD wollte ursprünglich eine eigene Befehlssatz-Erweiterung entwickeln, die der „Bulldozer“-Architektur hätte mitgegeben werden sollen und dieser ein Alleinstellungsmerkmal beschert hätte. Dazu kam es aber nicht – weshalb, ist bis heute unklar. Ein Großteil der Befehle ist stattdessen in anderen Erweiterungen aufgegangen.

Statt SSE ging es ab dem Jahr 2008 mit AES weiter. Der „Advanced Encryption Standard“ (AES) um-

fasst sieben Instruktionen, welche kryptografische Operationen beschleunigen. Truecrypt unterstützt AES und kann im Idealfall bis zu viermal schneller arbeiten. Meist limitieren aber andere Komponenten, etwa die Festplatte. Intels Core-i5-Modelle auf Clarkdale-Basis, die Gulftown-Sechskerner und alle i5-/i7-CPUs ab der Sandy-Bridge-Architektur beherrschen den Standard, bei AMD sind alle CPUs nach der Bulldozer-Architektur (also Piledriver, Steamroller, Jaguar usw.) AES-fähig. Mittlerweile beherrschen sogar einige ARM-Prozessoren AES, wovon vor allem NAS-Systeme und kleine Server profitieren können.

## AVX ...

Grundlegende Neuerungen folgten 2011. Sandy Bridge und Bulldozer führten erneut einen neuen Befehlssatz ein, der die Registerbreite ein weiteres Mal auf nunmehr 256 Bit erhöhte. Das erlaubte die parallele Verarbeitung von beispielsweise vier Werten mit 64 Bit (Fließkomma mit doppelter Genauigkeit) oder 8 mit 32 Bit (einfache Genauigkeit).

Zudem enthält AVX auch nicht-destruktive Befehle: Werden beispielsweise Werte aus den Registern A und B addiert, kann das Ergebnis in Register C abgelegt werden. Bei SSE überschreibt es dagegen Register A und für weitere Berechnungen muss der ursprüngliche Wert gegebenenfalls erneut aus dem L1-Cache in die, für die eigentlichen Rechenoperationen genutzten, Register geladen werden.

Bulldozer unterstützt zusätzlich exklusiv die aus AMDs SSE5-Vorschlag hervorgegangenen Befehlssätze XOP und CVT 16 und zusätzlich FMA 4. Einen Teil von deren Funktionalität hat Intel in Form der „Advanced Vector Extensions 2“ mit der Haswell-Architektur eingeführt, wodurch es zu einiger Verwirrung, kam. Nicht gerade durchsichtiger wurde die Lage dadurch, dass Celeron- und Pentium-Modelle anfangs überhaupt keinen AVX-Support boten, dann schon und heute wieder nicht.

Wie alle Befehlssatz-Erweiterungen benötigt auch AVX-Unterstützung durch die Software. Betriebssystem-seitig hat Microsoft erst ab Windows 7 SP1 eine entsprechende Unterstützung integriert. Erhielten die SSE-Register bei ihrer Einführung die Bezeichnung „XMM“, so wurden die AVX-Register nun als „YMM“ in Datenblättern und Anleitungen bezeichnet.

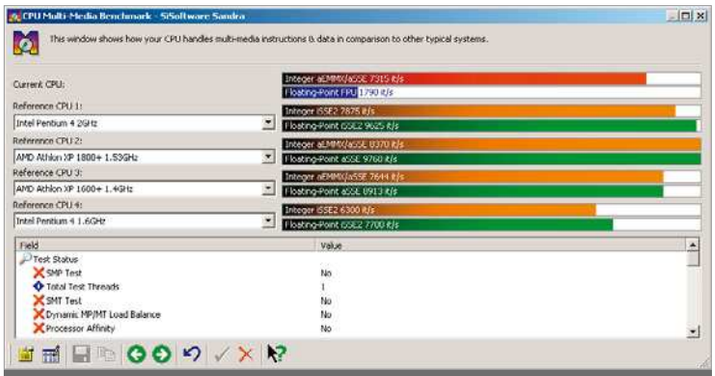
## ... und AVX 2

Die nächste logische Entwicklungsstufe hieß AVX 2 und wurde mit den Haswell-Prozessoren 2013 eingeführt – als Codename wählte Intel „HNI“ für „Haswell New Instructions“. Die AVX-2-Befehle ermöglichen unter anderem schnellere, gezielte Manipulationen und Abfragen von einzelnen Bits in einem längeren String. Intel verspricht die Beschleunigung von diversen Multimedia-Anwendungen, insbesondere Spielen, Audio- und Video-Bearbeitung.

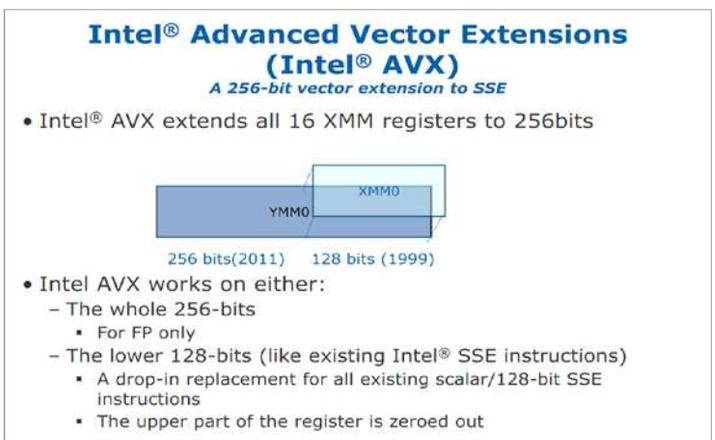
Die neuen Befehle ermöglichen es unter anderem, Registerinhalte direkt als Maske zu nutzen und erlauben schnellere, gezielte Manipulationen und Abfragen von einzelnen Bits aus längeren Zeichenketten („Strings“). Etwas greifbarer sind dagegen die prinzipiellen Erweiterungen, die ähnliche Wirkung entfalten sollen: 256 Bit Integer und FMA 3.

## 256-Bit-Integer-Kommandos

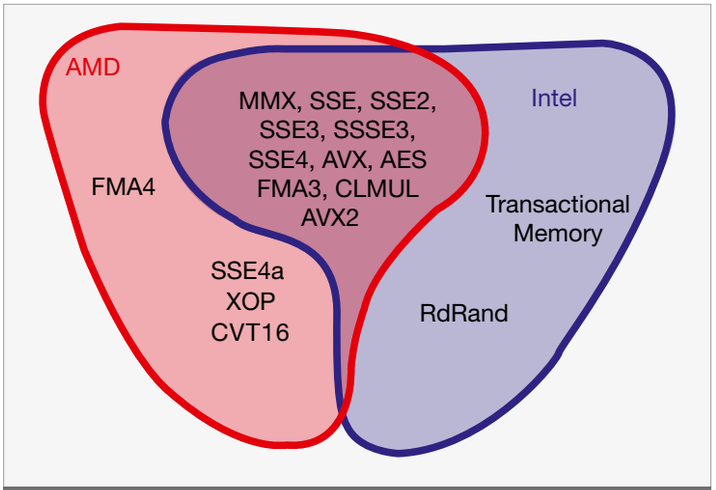
AVX (1)-CPUs arbeiteten zwar theoretisch mit 256 Bit Registern,




Mit synthetischen Benchmarks ist es möglich, den Einfluss von SIMD-Befehlen auf die Leistung zu testen und zu vergleichen.




Die mit AVX eingeführten Register YMM0 bis YMM15 lassen sich nur für Gleitkommaberechnungen nutzen. Für Integer-Berechnungen stehen bis zur AVX2-Einführung nur die alten SSE-Befehle zur Verfügung, welche die untere Hälfte wie die mit SSE1 definierten Register XMM0 bis XMM15 ansprechen.




Mehr als ein Dutzend Erweiterungen des Befehlssatzes sind in aktuellen Prozessoren implementiert. Leider sind nicht alle kompatibel, nur die Schnittmenge (Bildmitte) wird sowohl von AMD als auch Intel unterstützt.

Bild: PC Games/Hello Games

Trotz aller Fortschritte führen Befehlssatz-Erweiterungen immer wieder zu Problemen: So stürzte beispielsweise No Man's Sky ab, wenn der Prozessor kein SSE 4 beherrschte (mittlerweile per Patch behoben).

dies kam allerdings eher selten vor. Denn bis dato nutzten lediglich eine Reihe von Gleitkomma-Befehlen die erweiterten Register. Integer-Werte verwendeten nur die untere Hälfte der Register, der Rest wurde mit Nullen gefüllt. AVX 2 behebt diese Beschränkung auf alte SSE-Befehle und erweitert den Befehlssatz um entsprechende 256-Bit-Integer-Kommandos.

## FMA 3 und FMA 4

FMA 3 haben wir bislang übergangen. „Fused Multiply-Add"-Einheiten können in einem Arbeitsschritt eine Multiplikation und eine Addition/Subtraktion durchführen. Dies führt einerseits zu einem Genauigkeitsgewinn, da nur einmal am Ende der Operation gerundet wird. Andererseits beschleunigt die Kombination auch den Rechenablauf und damit die Gesamtleistung.

Die „3" steht hier aber nicht für die dritte Version dieser Befehlssatz-Erweiterung, sondern für die Tatsache, dass drei Register verwendet werden. Eine „fusionierte" Multiplikation ergibt also in einem Rechenschritt das Ergebnis einer Operation wie (A * B) + C. Das Ergebnis überschreibt allerdings eines der Quellregister.

Ein Mangel, der natürlich umgehend mit „FMA 4" behoben werden musste. Analog zu FMA 3 gibt die „4" bei der Nachfolge-Erweiterung ebenfalls die Zahl der verwendeten Register an: (A * B) + C = D; nach der Verrechnung der Werte aus drei Registern wird das Ergebnis also in einem vierten gespeichert.

Beide Entwicklungen stammen von AMD – Intel hat zwar FMA3 seit der Haswell-Generation implementiert, für FMA 4 liegen aber bislang noch keine Informationen vor. Bevor Intel sich auf FMA3 festlegte, hatte AMD bereits angekündigt, in zukünftigen CPU-Architekturen eventuell beide Varianten zu implementieren.

Für Bulldozer-Besitzer könnte die Entscheidung Intels dagegen bedeuten, dass die FMA4-Fähigkeit ihrer SIMD-Einheiten nie genutzt werden wird, da Softwareentwickler eine flächendeckende FMA3-Unterstützung erwarten können, während eine FMA4-Fixierung nur das obere Ende von AMDs letztjährigen Neuvorstellungen betrifft.

## AVX 3

Die vorerst letzte Erweiterung AVX 3 (auch bekannt als „AVX-512") ist zwar schon spezifiziert und seit 2013 in Grundzügen bekannt; sie war lange Zeit aber lediglich auf Beschleunigerkarten von Intel („Xeon Phi", Codename: „Knights Landing") verfügbar. Erst in Gestalt des Skylake-X-Prozessors bekam AVX-512 (F) in jüngster Vergangenheit eine minimale Verbreitung.

## Befehlssatzerweiterungen heute

Die Erweiterungen des ursprünglichen x86-Befehlssatzes sind eine Erfolgsstory. Mit jeder neuen Version wurden Schwächen und Unzulänglichkeiten der ursprünglichen x86-Architektur beseitigt. Zwar dauert es jeweils einige Jahre, bis die Verbreitung der CPUs so hoch ist, dass sich die Entwickler trauen, ihre Software entsprechend zu compilieren; doch der Leistungsgewinn ist im Regelfall enorm.

Außerhalb des x86-Universums hören die Erweiterungen auf die Namen Altivec, VMX, Velocity Engine (Motorola-CPUs in den alten Apple-Power-PCs) oder VMX128 in IBMs Xenon-PowerPC-CPU, in der Xbox 360 oder dem IBM Cell in der Playstation 3. Altivec galt seinen Pendants aus der Intel-Welt lange Zeit sogar als deutlich überlegen.

# So funktionieren Multiprocessing und Multitasking

Die Verbesserungen, die wir bisher beschrieben haben, betreffen nahezu alle den CPU-Kern und dessen direkte Rechenleistung. Pipelining, Out-of-Order-Execution, Befehlssatzerweiterungen und so weiter sind jedoch nicht beliebig ausbaubar.

## Entwicklungs-Barrieren

Mehrere Grenzen tun sich für CPU-Entwickler auf: Einerseits ist der Platz auf dem Siliziumplättchen nicht beliebig erweiterbar. Jeder CPU-Generation steht nur eine gewisse Anzahl an Transistoren zur Verfügung; dieses so genannte „Transistorbudget“ wird zwar mit jeder neuen Generation größer, doch wird ein Großteil der mittlerweile sogar milliardenfach vorhandenen elektrischen Schalter schon alleine für Cache benötigt.

Andereseits ist der CPU-Takt, der zweifellos jeden Kern schneller machen würde, nicht beliebig steigerbar. In der Vergangenheit hat sich eine Taktgrenze von etwa vier bis fünf Gigahertz herauskristallisiert, je nach CPU-Architektur kann diese etwas weiter oben oder unten angesiedelt sein.

## Mehr Prozessoren

Einen recht einfachen Weg aus diesem Dilemma fanden die Entwickler schon vor Jahrzehnten: Sie kombinierten einfach mehrere Prozessoren zu einem „Superrechner“, was als „SMP“ (Symmetric Multiprocessing) bezeichnet wird.

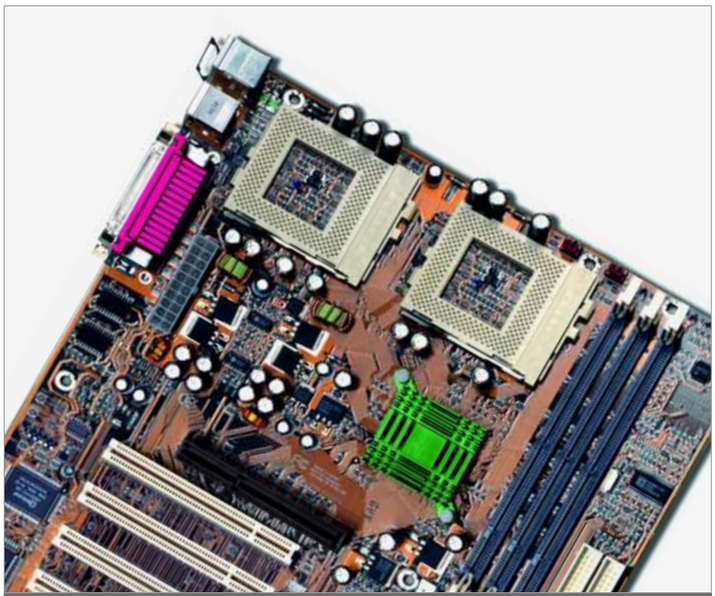
Tatsächlich basiert die Leistungsfähigkeit älterer wie neuerer Supercomputer meist nur auf der nackten Anzahl an Prozessoren im System. Eine Cray X-MP von 1983 vereinte (damals bescheidene) vier CPUs, aktuelle Vertreter der Supercomputerliste arbeiten mit knapp einer halben Million Prozessoren! Warum sollte man also nicht einfach mehrere Prozessoren auch in einem Desktop-PC nutzen? Bevor wir uns dieser Frage näher widmen, müssen wir uns aber ansehen, wie ein Programm genau aufgebaut ist.

## Prozesse und Threads

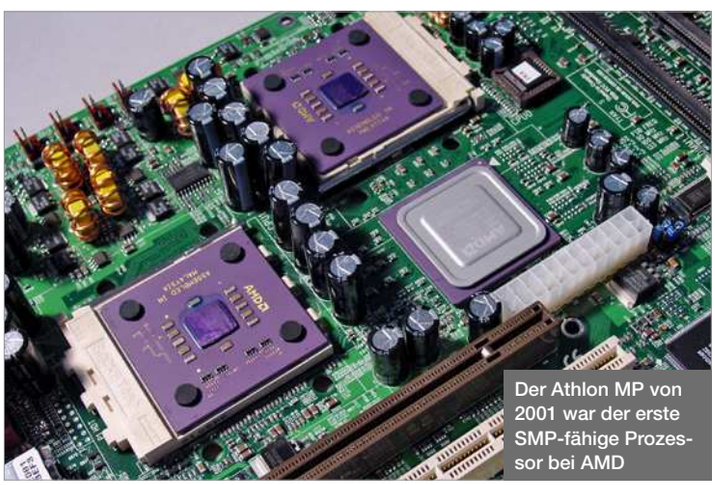
Wenn wir auf unserem Computer ein Programm starten, öffnen wir damit in der Regel einen Prozess. Früher, zu DOS-Zeiten, bestand ein Prozess immer aus genau einem Thread. Bei vielen Kommandozeilen-Anwendungen ist das auch heute noch so. Wenn das Programm gestartet wird, sind keine Ein- oder Ausgaben mehr möglich, ohne die Laufzeit des Programms zu unterbrechen.

Windows-Programme oder allgemein Programme mit Benutzeroberfläche bestehen dagegen zumindest aus zwei Threads: der Oberfläche und dem eigentlichen Arbeitsthread. So ist es möglich, eine Oberfläche auch dann noch zu bedienen, wenn das Programm auf Vollgas werkelt, bei einer Videosoftware zum Beispiel während ein Film umgewandelt wird.

Eine andere Möglichkeit ist es, Oberfläche und Programm strikt zu trennen. In einem solchen Fall öffnet die Oberfläche beim Start der Videoumwandlung ein zusätzliches Programm, meist auf Kommandozeilenebene. Ein typischer Vertreter dieser Gattung ist der MP3-Encoder Lame. Von dieser Möglichkeit wird aber nur selten Gebaruch gemacht. Wie genau Prozesse und Threads entworfen wurden, fällt in das Aufgabengebiet der Programmierer



Zur Legende wurde Abits BP6, das als eines der ersten Mainboards den Einsatz zweier günstiger Prozessoren in einem Desktop-PC ermöglichte



Der Athlon MP von 2001 war der erste SMP-fähige Prozessor bei AMD

## Worin sich Prozess, Thread und Task unterscheiden

**Die Begriffe Task, Thread und Prozess sind verwirrend und werden häufig selbst von Experten durcheinandergebracht. Daher hier ein kurzer Überblick, der hoffentlich die Unterscheidung erleichtert.**

### Prozess

**Der Prozess lässt sich am einfachsten von allen anderen Vorgängen abgrenzen. Er ist pro Programm immer nur einmal vorhanden – ein Programm eröffnet also immer genau ein Prozess. Dieser Prozess bekommt seinen eigenen Speicherbereich zugewiesen, er ist also unabhängig von etwaigen anderen Prozessen. Auf einem SMP-System lassen sich immer mindestens zwei Prozesse parallel betreiben.**

### Thread

**Ein Thread wird gelegentlich auch als Programmfaden bezeichnet. Er ist also Teil eines Prozesses. Als solcher kann er nur auf den Speicherbereich (korrekt: den Adressraum) seines Mutterprozesses zugreifen. Auch alle anderen wichtigen Ressourcen, etwa die CPU-Zeit, hängen vom Prozess ab. Ein Prozess hat immer mindestens einen Thread.**

### Task

**Nun können wir uns auch um den letzten verbliebenen Begriff kümmern. Nach moderner Definition ist ein Task ein Prozess mit mehreren Threads – ein Prozess demnach aber immer nur ein Prozess mit genau einem Thread.**

**Häufig wird der Task umgangssprachlich aber einfach mit einem Programm gleichgesetzt – egal mit wie vielen Threads. Selbst Microsoft redet vom „Task-Manager“ und nicht etwa vom „Prozess-Manager“.**

und der Compiler. Daneben ist es auch möglich, einen Prozess nicht nur auf zwei, sondern auf *mehrere* unabhängige Threads aufzuteilen – so wie im oben erwähnten Windows-Programm, nur dass jetzt nicht mehr lediglich Oberfläche und „Arbeits-Thread“ getrennt werden, sondern die komplette Ausführung in mehrere, möglichst unabhängige Threads zerlegt wird.

### Arbeitsteilung

Ist ein Programm erst einmal so aufgeteilt, lassen sich die Prozesse auch auf mehrere Prozessoren verteilen, beispielsweise bei einer sehr ungleichen Verteilung der Rechenzeit zweier Aufgaben. Das macht die Rechner natürlich viel flexibler und leistungsfähiger.

Vorteil dieses Multi-Processings oder Multi-Threadings ist also entweder ein deutliches Geschwindigkeitsplus oder ein besseres Ansprechverhalten, wenn nur ein Prozessor mit einer Aufgabe ausgelastet ist und der zweite nach wie vor vollständig zum Arbeiten zur Verfügung steht. Wenn es gelingt, die Aufgaben gut zu verteilen, sind die Leistungssteigerungen immens: Die Rechenleistung des Systems steigt im Idealfall nahezu linear mit der Anzahl der Prozessoren, und das bei mittlerweile recht geringen Mehrkosten.

Doch ein Problem gibt es: Bei vielen SMP-Systemen wird die komplette externe Peripherie geteilt. Das kann sich als Bremsklotz erweisen. Chipsatz, Speicher und Grafikkarte sind dann nur einfach vorhanden, was früher den PCI-Bus überforderte und auch heute noch für manchen Engpass sorgen kann (mehr dazu im Mainboard-Kapitel).

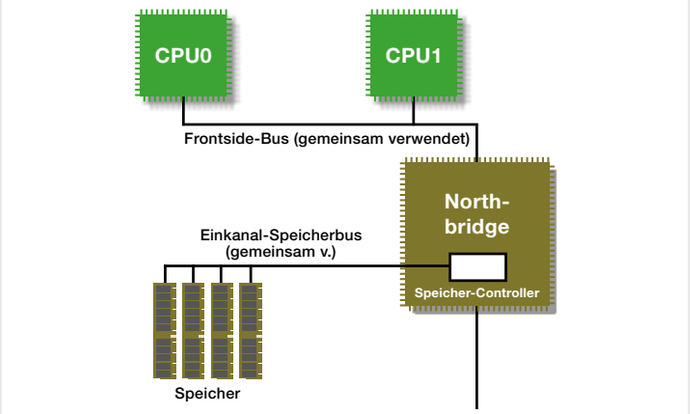
### Betriebssysteme und CPUs

Softwareseitig sind in der x86-Welt die Voraussetzungen für den Einsatz mehrerer Prozessoren seit Windows NT von 1993 geschaffen.

## Wie Multiprocessing in der x86-Welt umgesetzt wurde

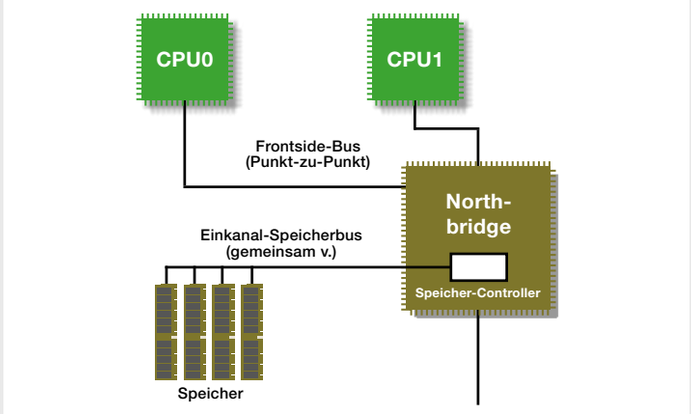
**Seit Windows eine echte Unterstützung für SMP mitbrachte, gab es immer wieder Ansätze, das Rechnen mit mehreren Prozessoren umzusetzen. DIe wichtigsten stellen wir in dieser Übersicht vor.**

### Überblick: Pentium-III-System (SMP)




In den anfänglichen SMP-Designs, etwa der Pentium-III-Plattform, teilten sich zwei Prozessoren einen gemeinsamen Frontside-Bus.

### Überblick: Athlon-MP-System (SMP)




AMD stattete seine Systeme mit zwei Athlon-MP-Prozessoren aus, wobei jede CPU über einen eigenen Punkt-zu-Punkt-FSB verfügen konnte.

Allerdings war ein entsprechendes System für Heimanwender lange Zeit schlicht zu teuer und zu wenig praxistauglich, so dass SMP erst mit Windows 2000 und XP mehr ins Bewusstsein der Anwender rückte. Hardwareseitig wurden zudem bestimmte Prozessorfunktionen benötigt, die erst mit dem 386 Einzug hielten und dadurch den SMP-Aufbruch in der x86-Welt ebenfalls verzögerten.

## Präemptives Multitasking

Natürlich konnte man aber spätestens seit Windows 95 auch auf einem Ein-Prozessor-System mehr als nur ein Programm ausführen. Und dass dies möglich wurde, verdanken wir einer Funktion namens „Multitasking“.

Die scheinbar parallele Verarbeitung mehrerer Programme darf allerdings nicht darüber hinwegtäuschen, dass die Arbeit tatsächlich immer abwechselnd ausgeführt wird. Ein so genannter „Scheduler“ teilt der Reihe nach jeder Anwendung abhängig von ihrer Priorität eine gewisse Prozessorzeit zu – mit Betonung auf „der Reihe nach“.

Effektiv in Bearbeitung ist auf einem „traditionellen“ Prozessor immer nur ein Thread! Der Anwender hat lediglich den Eindruck, mehrere Programme würden tatsächlich gleichzeitig arbeiten, da der Wechsel zwischen den Tasks sehr schnell vonstatten geht. Aufgrund der Arbeitsweise bezeichnet man dies auch als „präemptives Multitasking“, abgeleitet vom englischen „preempt“, was „jemandem zuvorkommen, um ihn von etwas abzuhalten“ bedeutet.

Präemptives Multitasking ist übrigens keine Errungenschaft der x86-Welt, respektive Microsofts, Intels oder AMDs. Im Gegenteil: Lange Zeit war diese Technik ein Privileg anderer Architekturen wie Motorolas 68000er-Reihe samt passender

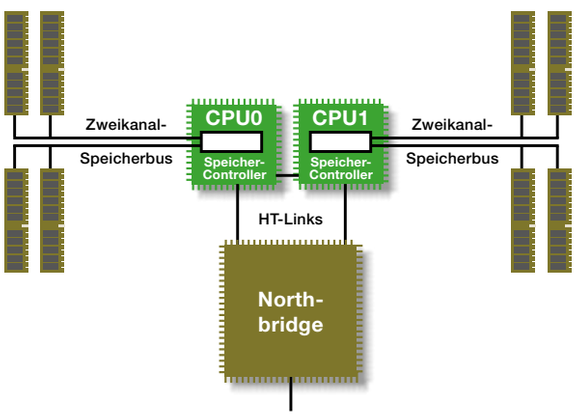
Bild: © Bill Bertram 2006, CC-BY-2.5 via Wikipedia

Commodores Amiga brachte schon 1985 präemptives Multitasking in beeindruckender Qualität – lang bevor dies auf x86-PCs sinnvoll nutzbar wurde

| Name | PID | Status | Benutzerna... | CPU | Arbeitssp... | Threads | Beschreibung |
|---|---|---|---|---|---|---|---|
| aavus.exe | 2292 | Wird ausgeführt | SYSTEM | 00 | 112 K | 2 | AAV - Online Update Dienst |
| AcroRd32.exe | 4560 | Wird ausgeführt | GMV | 00 | 6.864 K | 9 | Adobe Acrobat Reader DC |
| AcroRd32.exe | 17272 | Wird ausgeführt | GMV | 00 | 177.588 K | 19 | Adobe Acrobat Reader DC |
| Adobe CEF Helper.exe | 5244 | Wird ausgeführt | GMV | 00 | 2.268 K | 12 | Adobe CEF Helper |
| Adobe Desktop Servi... | 14112 | Wird ausgeführt | GMV | 00 | 9.408 K | 27 | Creative Cloud |
| AdobeIPCBroker.exe | 7768 | Wird ausgeführt | GMV | 00 | 1.864 K | 16 | Adobe IPC Broker |
| AdobeUpdateService... | 952 | Wird ausgeführt | SYSTEM | 00 | 104 K | 5 | Adobe Update Service |
| Agent.exe | 2448 | Wird ausgeführt | SYSTEM | 00 | 1.240 K | 47 | EaseUS Todo Backup Agent A. |
| | | | | | 236 K | 5 | Adobe Genuine Software Inte. |
| | | | | | 2.680 K | 4 | Application Frame Host |
| | | | | | 64 K | 2 | Adobe Acrobat Update Servic |
| | | | | | 4.580 K | 6 | Windows Graphisolierung für |
| | | | | | 3.020 K | 31 | Adobe Bridge |
| | | | | | 444 K | 25 | Calculator.exe |

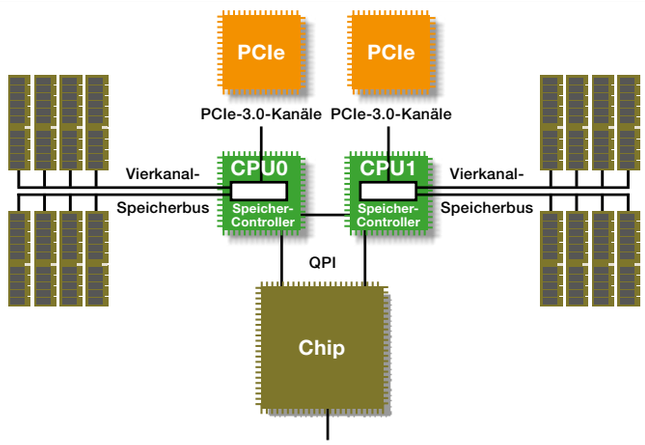
Wie viele Threads ein Prozess aufgemacht hat, lässt sich am leichtesten mit dem Task-Manager herausfinden – Sie müssen die Spalte aber manuell hinzufügen

### Überblick: Opteron-System (SMP)




Die Opteron-Architektur sah für jede CPU einen eigenen Speicherbus vor. Zudem sind CPUs per HT-Link untereinander sowie mit dem Chipsatz verbunden.

### Überblick: Xeon-System (SMP)




Und noch ein aktuelles Xeon-ES-System: Hier ist der Speicher mit vier Kanälen angebunden. Außerdem verfügen die CPUs über integrierte PCI-Express-Controller.

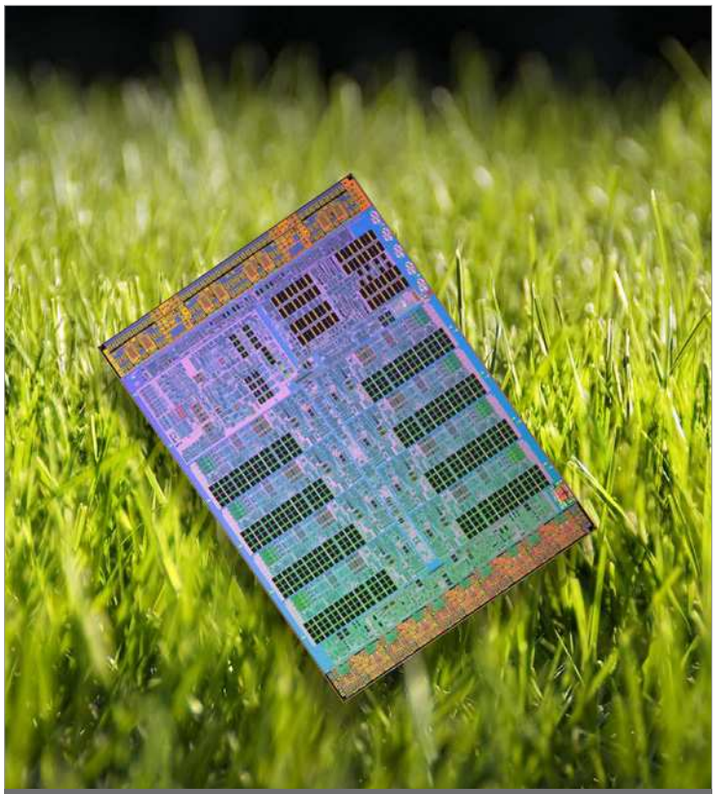
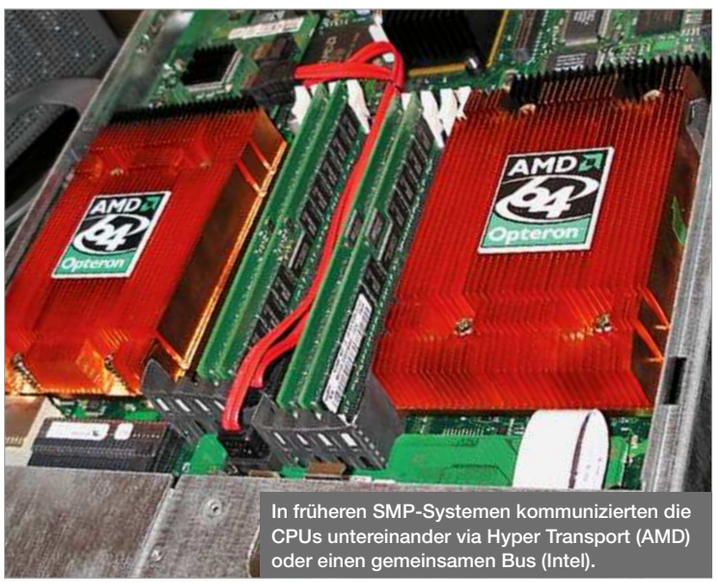


Bild: IBM

IBMs Cell-Processor basiert auf einer der wenigen CPU-Architekturen, die mit asymmetrischem Multiprocessing arbeiten. Gut zu erkennen im Bild ist die „Master"-Einheit oben sowie die acht „einfachen" Recheneinheiten links und rechts




In früheren SMP-Systemen kommunizierten die CPUs untereinander via Hyper Transport (AMD) oder einen gemeinsamen Bus (Intel).

Betriebssysteme, die zum Beispiel in den legendären Amiga-Rechnern der 1990er-Jahre zum Einsatz kamen. Die Implementierung der Technik war ihrer Zeit damals um Jahre, wenn nicht Jahrzehnte voraus – auch heute noch kann man sich auf Online-Videoplattformen davon überzeugen, wie flüssig das Multitasking damals vonstatten ging (beispielsweise unter https://youtu.be/emLXJadj9s4 ).

## Symmetrisch und asymmetrisch

Doch zurück zum Multiprocessing: Das eben vorgestellte Multitasking steht keinesfalls im Widerspruch zu Multiprocessing. Im Gegenteil: Ist der Scheduler erst mal auf eine möglichst gleichmäßige Aufteilung der Arbeit eingestellt, kann man diese auch leicht auf mehrere Prozessoren verteilen. Weil die Arbeit in der Regel gleichmäßig auf die verfügbaren Ressourcen verteilt wird, spricht man auch von symmetrischem Multiprocessing.

Beim asymmetrischen Multiprocessing dagegen obliegt die Verteilung der Arbeit einem einzelnen Prozessor. Fällt dieser „Master" aus, kann die Maschine nicht mehr weiterrechnen. Ein Beispiel für eine solche Architektur ist IBMs Cell-Prozessor, der in der Spielekonsole Playstation 3 zum Einsatz kam. Der Chip bestand in der Ur-Ausführung aus einem „Power-PC-Processing-Element" (PPE) und acht „Synergistic Processing Elements" (SPE). Asymmetrische Lösungen waren wegen der anspruchsvollen Programmierung lange Zeit eher unpopulär, erlebten aber mit der Integration von Grafikprozessoren in die Gesamtleistung eines PCs ein Comeback .

## SMP-Probleme

Bei aller Euphorie haben SMP-Architekturen eine Reihe handfester Nachteile und Probleme. Ein grundsätzliches Problem ist die Anzahl der begrenzten externen Ressourcen.

Wichtige PC-Komponenten abseits des Prozessors sind ja nach wie vor nur einmal vorhanden, etwa Speicher, Grafikkarte oder Massenspeicher. Wollen mehrere Prozessoren auf dieselbe Ressource zugreifen, kommt es zu Konflikten, aus denen schlimmstenfalls Stillstände resultieren, die so genannten stalls (engl.: to stall = zum Stillstand bringen, stillstehen). Je mehr Prozessoren versuchen, simultan zum Beispiel auf die PCI-Express-Grafikkarte zuzugreifen, desto mehr Rechenzeit wird mit Stillstand vergeudet.

## Speicher-Probleme

Der gleiche Effekt tritt auch beim Arbeitsspeicher auf. Hierfür existiert das sogenannte Locking (to lock = sperren, am Zugang hindern). Multiprozessor-Kernel von Betriebssystemen müssen einige zusätzliche Befehle implementieren, die beispielsweise den Arbeitsspeicher vor parallelen Zugriffen schützen, um die Datenkonsistenz zu wahren, sprich: um Datensalat zu vermeiden. Greift eine CPU in einem SMP-System auf den Arbeitsspeicher zu, so wird der Zugriff auf diesen zunächst für alle weiteren vorhandenen Prozessoren gesperrt und dem wartenden Prozessor der exklusive Zugriff gewährt. Anschließend muss der Arbeitsspeicher wieder für alle Prozessoren freigegeben werden.

Das Problem an diesem Prozedere ist, dass der dabei entstehende Overhead gigantisch ist. Dieser Vorgang, der den ohnehin schon ausbremsenden Zugriff auf den Speicher noch weiter verlangsamt, wirkt nicht sonderlich leistungsfördernd. Das Problematische an diesem Locking ist die Tatsache, dass nicht nur Single-threaded-Ap-

plications, also Programme, die nicht für den Betrieb auf mehreren Prozessoren konzipiert sind, dieses Locking verursachen, sondern auch Multi-threaded-Applications. Aus diesem Grund kann die Rechnung „Leistungszuwachs um den Faktor der CPU-Anzahl“ in der Praxis niemals aufgehen. Bei ausschließlichem Einsatz von nicht SMP-optimierter Software kann ein Programm sogar langsamer arbeiten als der gleiche Rechner mit nur einer CPU.

## Cache-Probleme

Ein weiteres Problem stellt die Kohärenz der Inhalte der unterschiedlichen Cache-Hierarchien dar. Cache-Kohärenz bedeutet die Wahrung der Gültigkeit der Inhalte der Caches. Wird beispielsweise ein Wert aus dem Arbeitsspeicher von beiden Prozessoren in den jeweils eigenen Cache geladen und von einem der Prozessoren anschließend (verändert) wieder zurückgeschrieben, muss der zweite Prozessor von diesem Vorgang etwas mitbekommen, um seine eigenen Caches entsprechend zu aktualisieren.

SMP-Systeme haben hierfür ein sogenanntes Cache-Kohärenz-Protokoll, welches für die Korrektheit der Daten sorgt. Das traditionelle Protokoll dafür ist das sogenannte MESI-Protokoll. Jeder der Buchstaben steht hierbei für einen bestimmten Zustand einer Cache-Line (siehe Extrakasten rechts). Die Erhaltung der Korrektheit der Daten kostet jedoch ein klein wenig Performance, was je nach Umsetzung des Protokolls unterschiedlich gravierend ausfallen kann.

## Cache-Protokolle

Eine Erweiterung des MESI-Protokolls stellt das MOESI-Protokoll dar. Es ist zum MESI-Protokoll voll kompatibel, kennt jedoch den zusätzlichen Status „Owner“. Dieser zeigt, in welchem Cache welches Prozessors sich die aktuell gültigen Daten befinden. Somit können andere Prozessoren diese bei Bedarf aus dem Cache anfordern und sparen sich den Hauptspeicherzugriff.

## MOESI-Funktion

Hierbei passiert Folgendes: Stellt CPU 1 eine Leseanforderung (Read-Request) auf einen bestimmten Speicherbereich, so wird dieser in der Regel direkt aus dem Arbeitsspeicher in den Cache der CPU geladen. Merkt die interne Logik jedoch, dass eine weitere CPU genau diesen Speicherbereich bereits im Cache hat, so wird der Read-Request vom langsamen Arbeitsspeicher auf den schnellen Cache der betreffenden CPU umgeleitet. Somit werden massiv Taktzyklen eingespart. Auch der Speicherbus profitiert, da weniger Zugriffe auf den Speicherbus stattfinden und er somit für andere Aufgaben verfügbar ist.

## NUMA

Was bei zwei Kernen bzw. Prozessoren noch überschaubar erscheint, gerät bei acht Kernen bzw. Prozessoren zu einem aufwendigen und bremsenden Durcheinander. Ab einer gewissen Komplexität ist das System hauptsächlich damit beschäftigt, auf Cache-Abgleiche zu warten. Ein Ausweg ist das ccNUMA-Modell („Cache coherent Non-Uniform Memory Access“), in dem es einerseits für das Betriebssystem zwar einen gemeinsamen Speicherbereich gibt, andererseits aber versucht wird, Daten möglichst lokal zu halten, um Overhead damit schon im Keim zu ersticken.

## Software

Vor allem aber sind die Software-Entwickler gefragt. Diese müssen die Programmstrukturen so gestalten, dass mehrere Threads oder Prozesse unabhängig voneinander arbeiten. Dies ist aber gerade bei Spielen eine enorme Herausforderung.

# Welche Bedeutung das M(O)ESI-Cacheprotokoll hat

**Das MESI-Protokoll sorgt dafür, dass Daten im Arbeitsspeicher konsistent bleiben – andernfalls könnte es passieren, dass Prozessor 1 Daten liest, verrechnet und erneuert, während Prozessor 2 mit den alten, falschen Daten weiterrechnet. Dabei werden folgende Zustände unterschieden.**

### Modified

**Der Inhalt im Arbeitsspeicher wurde seit dem Spiegeln in den lokalen Prozessorcache modifiziert und stimmt somit nicht mehr überein.**

### Owner

**Ähnlich wie Modified mit dem Unterschied, dass es nicht nur eine, sondern mehrere Cache-Kopien gibt, die aktualisiert werden müssen.**

### Exclusive

**Der Inhalt im Arbeitsspeicher stimmt noch mit dem gespiegelten Inhalt im Cache überein. Die Cache-Line befindet sich nur im Cache eines einzelnen Prozessors.**

### Shared

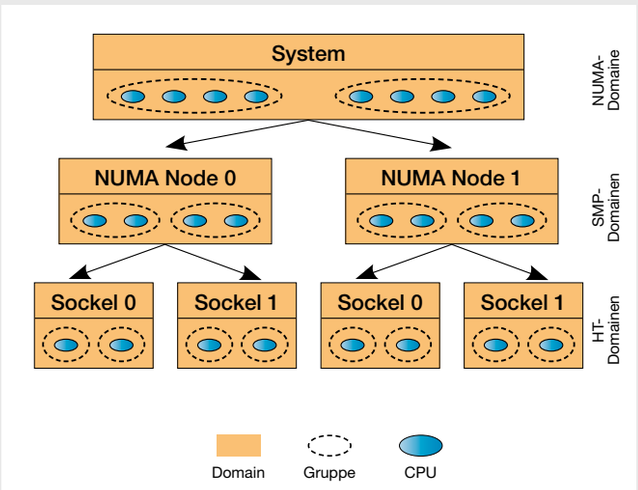
**Wie Exclusive – mit dem Unterschied, dass sich die Cache-Line in mehreren Caches befindet. Bei Aktualisierung der ursprünglichen Daten im Arbeitsspeicher werden alle Caches auf den neuesten Stand gebracht.**

### Invalid

**Der Inhalt im Prozessorcache ist ungültig, ein Zugriff erzeugt einen Cache-Miss. Die Daten müssen neu aus dem RAM geladen werden.**

# Wie die NUMA-Architektur aufgebaut ist

**Seit der Nehalem-Generation hat Intel mehrstufige NUMA-Strukturen bis hinab zu SMT integriert, um die Lokalität von Daten optimal nutzen zu können.**



# Von SMP zu SMT: Multi- und Hyper-Threading

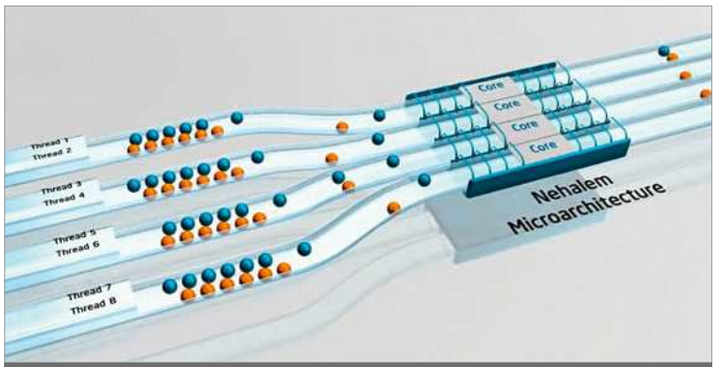



Bild: Intel

Hyper-Threading soll die Funktionseinheiten moderner CPUs besser auslasten. Bis heute klappt das im Alltag aber nicht immer

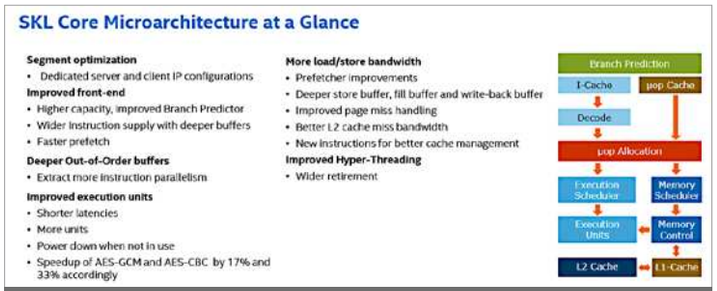
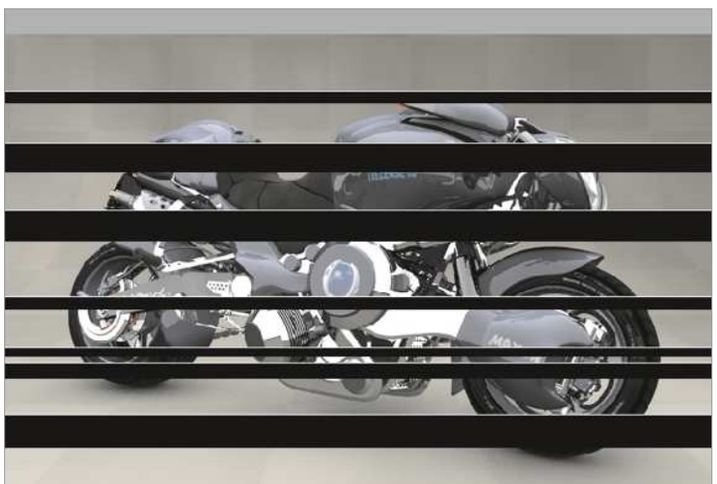



Bild: Intel

Auch wenn es nicht den Anschein hat: Intel optimiert Hyper-Threading kontinuierlich weiter. Zuletzt spendierten die Entwickler der Skylake-Architektur ein breiteres Frontend, wodurch auch das Leistungsplus wächst



Ein zeitgenössischer Benchmark war der Cinebench: Er zeigte, was theoretisch mit SMT möglich war – durch die Aufteilung in mehrere Threads konnten CPUs deutliche Leistungsvorsprünge verbuchen

Symmetric Multiprocessing (SMP) und Multitasking blieben nicht die einzigen Techniken, um Prozessoren besser auszulasten. Eine weitere ist das Simultaneous Multi Threading (SMT), welches versucht, mit möglichst wenigen zusätzlichen Ressourcen die Einheiten der CPU besser zu beschäftigen.

## Multi-Threading

SMT ist nicht neu; schon für Alpha-Prozessoren wurde diese Technik in den 1990er-Jahren entwickelt, kam dort aber nicht mehr zum Einsatz – die Entwicklung des Prozessors wurde vorher eingestellt. So wurde SMT im Wesentlichen durch Intels Pentium 4 bekannt, der ab 2002 entsprechende Einheiten mitbrachte. Vermarktet wurde das Feature unter dem Namen „Hyper-Threading (HT)".

Der Clou an HT bzw. SMT ist nun, die Vorteile von SMP-Systemen auf nur einen Prozessor zu übertragen, zumindest teilweise. Bei aktiviertem HT gaukelt der Prozessor dem System einfach zwei CPUs vor, obwohl physisch nur eine vorhanden ist. Das Betriebssystem glaubt, zwei Prozessoren gefunden zu haben und verwaltet das System wie ein echtes Mehr-Prozessorsystem. Mit optimiertem Code können sich drastische Leistungssteigerungen einstellen; bei Single-Threaded-Software passiert jedoch gar nichts, in Einzelfällen arbeiten Hyper-Threading-CPUs – genau wie SMP-Systeme – sogar langsamer.

Um einem Betriebssystem zwei logische CPUs bereitstellen zu können, lassen die Entwickler verschiedene Bauteile im Kern auf verschiedene Arten teilen (shared), replizieren (replicated) und splitten (partitioned). Der Level-1- und -2-Cache dient beiden logischen Prozessoren (LPs) gleichermaßen als Puffer. Das hat den Vorteil, dass Daten, die in beiden LPs verarbeitet werden müssen, schnell und bequem erreichbar sind. Auf echten Multi-Prozessorsystemen mit separaten Caches für jede CPU wird das System ausgebremst, wenn Code verarbeitet wird, dessen Daten für beide Prozessoren relevant sind, wie wir es im vorhergehenden Abschnitt ja ausführlich beschrieben haben.

Andere Bauteile dagegen, wie etwa der Re-Order-Buffer, werden parallel und geteilt verwendet. Andere, deren reale Bauteile ohnehin mehrfach vorhanden sind, werden kurzerhand repliziert. Darunter fällt zum Beispiel die Einheit für das Register-Renaming oder der Instruction Pointer in der I-Fetch-Unit.

## Software

Die Voraussetzungen, damit eine Software auch in der Praxis von SMT profitiert, sind dieselben wie beim SMP: Es müssen mindestens so viele Arbeitsprozesse bzw. Arbeitsthreads existieren, wie Prozessoren vorhanden sind. Das zu realisieren, fällt in das Aufgabengebiet der Programmierer und der Compiler. Wenn das nicht der Fall ist, steht ein Prozessor still, während der andere unter Volllast arbeitet. Außerdem bringt es nichts, einen bereits voll ausgelasteten Prozessor mit zusätzlichen (SMT-)Aufgaben zu beladen. In der Regel haben die Prozessoren aber genug „Leerlauf", um weitere Befehle auszuführen.

## Probleme …

Tatsächlich bemühten sich nach einer Anlaufphase nicht nur Anwendungs-, sondern auch Spie-

leentwickler, die virtuellen Kerne gewinnbringend einzusetzen. Dies führte bei entsprechendem Ressourcenaufwand zu vorzeigbaren Ergebnissen: Anno 1404 lief beispielsweise mit aktivem SMT spürbar flüssiger als ohne.

Weit häufiger ergab sich aber überhaupt kein Vorteil und – schlimmer noch – manchmal sogar eine deutliche Leistungseinbuße. Trotz des positiven Anno-Beispiels waren besonders Spiele lange Zeit von dieser Problematik betroffen und auch Optimierungsversuche über den Grafiktreiber seitens Ati (damals noch eigenständig) sowie Nvidia konnten keine Abhilfe schaffen.

## … und Lösungen

Eine Ursache war, dass die bei der Einführung von Multi-Threading verfügbaren Desktop-Betriebssysteme zu wenig mit den zusätzlichen Ressourcen anzufangen wussten. Im Jahr 2002 kümmerte sich zumindest kein Desktop-Windows-Scheduler um die optimale Verteilung mehrerer Prozesse. So kam es häufig vor, dass sich ressourcenhungrige Threads in einem Prozess um die geteilten oder partitionierten Einheiten „streiten“ mussten, während andere Prozesse noch genügend Reserven hatten. Erst mit Windows 7 schuf Microsoft eine spürbare Verbesserung. Hier hielt ein neues Feature Einzug, das die Verteilung von Threads besser managen konnte und als „SMT Parking“ bekannt wurde.

## SMT Parking

Das aus „Core Parking“ hervorgegangene SMT Parking sorgt dafür, dass Threads nicht mehr ständig scheinbar wahllos zwischen den virtuellen Prozessoren hin- und herspringen, sondern versucht stattdessen, den Ressourcenhunger der Threads an die Leistungsfähigkeit der (virtuellen) Prozessorkerne anzupassen. Das klappt auch sehr gut und man kann SMT mittlerweile in der Regel aktiviert lassen.

Der Mechanismus funktioniert auch bei „echten“ Mehrprozessor-Systemen. Der Wechsel zu einem anderen Prozessor in einem anderen Sockel hat nämlich zur Folge, dass auch alle zugehörigen Threaddaten in den anderen Speichercontroller übertragen werden müssen, was Performance kosten kann. Dies wird durch SMT Parking unterbunden.

## Andere SMT-Lösungen

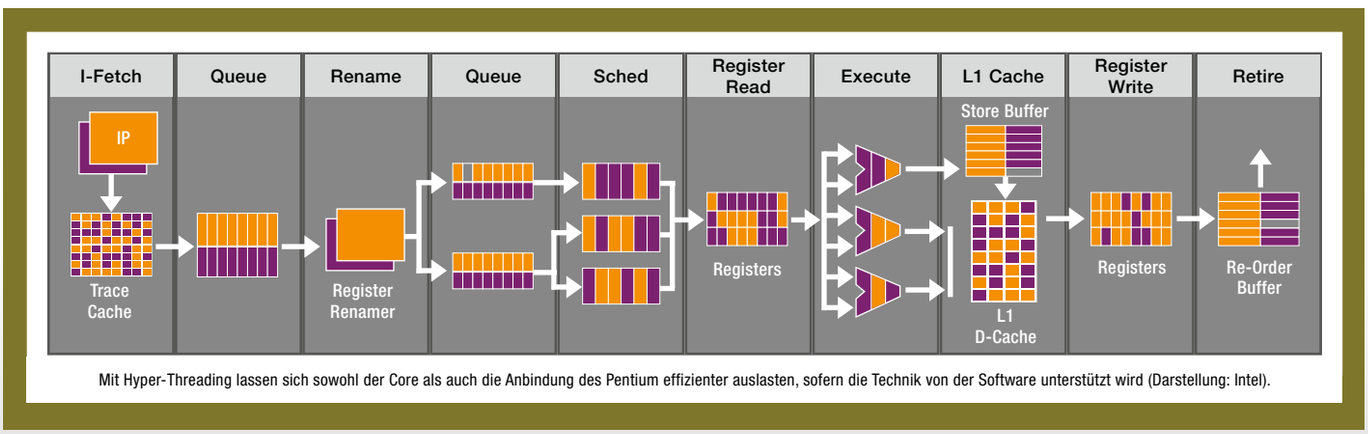
Ziemlich genau 15 Jahre lang gab es SMT bei x86-Prozessoren exklusiv bei Intel – AMD hatte keine entsprechende Technik im Angebot. Andere Prozessorarchitekturen aber sehr wohl! IBM beispielsweise hatte mit dem Power 5 eine entsprechende SMT-CPU im Programm.

Seit 2017 können aber auch Käufer einer AMD-CPU in den Genuss von Multithreading kommen. Verschiedene Einheiten müssen sich bei Ryzen und Co. die einzelnen Threads teilen – so zum Beispiel die Rename-Units, die TLBs oder die Load-Queue sowie natürlich die Ausführungseinheiten selbst. Andere, wie die Mikro-Op-, Store- oder Retirement-Queue, werden statisch aufgeteilt.

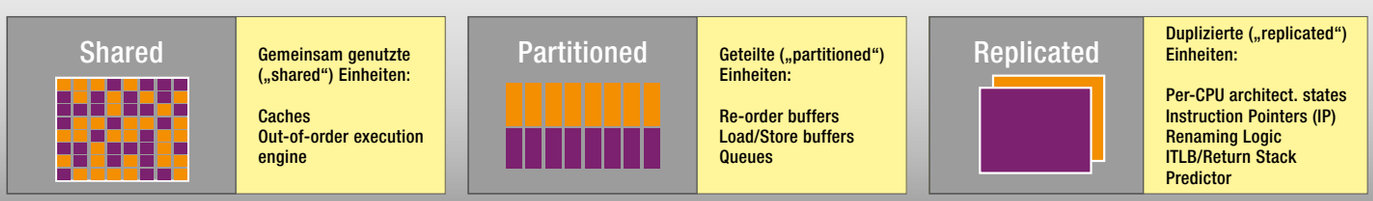
Bei der Verteilung der Ressourcen ermittelt ein CPU-Algorithmus, welcher Thread am „hungrigsten“ ist und gewährt diesem entsprechend Priorität – dies aber nur bei wirklich performance-kritischen CPU-Einheiten. In der Praxis funktioniert die AMD-Lösung sehr gut und bringt messbare Vorteile.

# Hyper-Threading am Beispiel des Intel P4

**Intel unterscheidet drei Arten, wie die Bauteile der CPU für zwei LPs verwaltet werden: shared, replicated und partitioned. Im Bild ist die ursprüngliche Aufteilung der Funktionseinheiten im Pentium 4 zu sehen; bei späteren Implementierungen stellte Intel weniger detaillierte Informationen zur Verfügung und bis heute ist unklar, wie genau die Technik bei Nehalem und späteren Architekturen umgesetzt wurde.**




Mit Hyper-Threading lassen sich sowohl der Core als auch die Anbindung des Pentium effizienter auslasten, sofern die Technik von der Software unterstützt wird (Darstellung: Intel).



# Das Beste aus beiden Welten: Mehrkern-Prozessoren

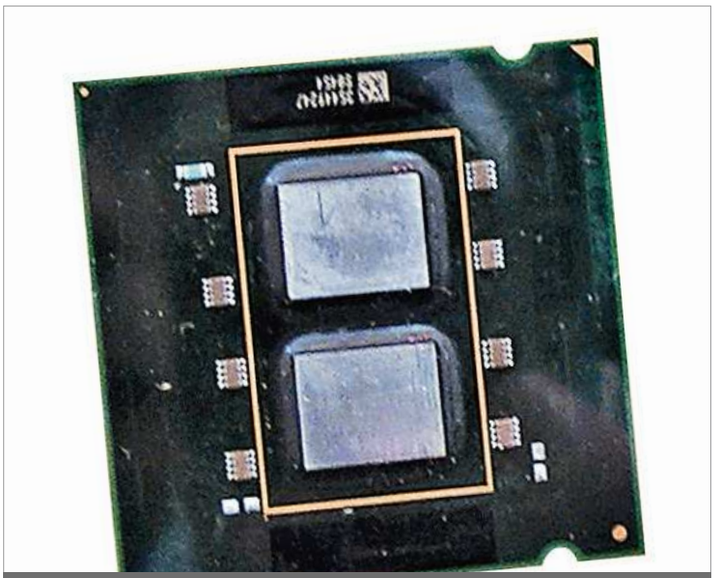


Bild: Intel

Intels erste Desktop-Version einer Mehrkern-CPU wirkte fast dilettantisch: Fachleute rümpften die Nase über die zwei Kerne, die scheinbar schlicht „aneinandergeklebt“ wurden (hier: Pentium 4 Presler)

Wahnsinn oder konsequente Weiterentwicklung? Intels Broadwell-E (Bild) bringt bis zu 10 Kerne (und zusätzlich Hyper-Threading!), Skylake-X gar 18.

Hyper Threading war ein gut gemeinter Versuch, die Auslastung moderner Prozessoren zu steigern und brachte in Einzelfällen, etwa beim Transkodieren von Filmen, tatsächlich ordentliche Leistungssteigerungen.

In der Regel ist SMT aber selbst bei aktuellen Prozessoren kaum mehr als ein nettes Zubrot: Schön zu haben, aber ohne durchschlagende Performance-Relevanz. Einen echten Durchbruch im Desktop-Bereich erlebte die Idee mehrerer Prozessoren erst mit den sogenannten „Multicore-CPUs“.

## Multicore-CPUs

Multicore-Prozessoren vereinen im Prinzip die Vorteile von SMP und SMT: viele Prozessorressourcen, gebündelt auf möglichst kleiner Chipfläche. Im Gegensatz zu SMP sind nicht länger mehrere Sockel nötig, selbst vier oder mehr Kerne lassen sich platzsparend und einigermaßen kühl auf einem gemeinsamen Siliziumplättchen betreiben – und solche CPUs passen zudem auf jedes handelsübliche Desktop-Board, ohne teure Zusatzkomponenten verbauen zu müssen.

Trotzdem bleibt die volle Rechenleistung erhalten, da alle performancekritischen Einheiten tatsächlich doppelt vorhanden sind: ALUs, Caches, Puffer; Sprungvorhersagen usw. sind alle ohne Einschränkungen für Prozesse und Threads nutzbar.

Möglich wurde das durch immer feinere Fertigungsprozesse, die Transistorzahlen im Milliardenbereich bei gleichzeitig moderater Wärmeentwicklung erlaubten. Mittlerweile ist es sogar üblich, bei Multicore-Prozessoren zusätzlich SMT einzusetzen – wenngleich dies meist den teureren Modellen vorbehalten bleibt.

Aktueller Stand sind vier bis acht Kerne pro Prozessorsilizium, bei dem offenbar der beste Kompromiss aus Performance-Gewinn und Transistorzahl zu liegen scheint – und das übrigens nicht nur im x86-Lager. Auch ARM-Prozessoren, wie wir sie in Smartphones, NAS-Servern oder Tablets finden, arbeiten in der Regel mit einer ähnlichen Kernzahl pro Chip.

Im Oberklassesegment sind aktuell aber auch bis zu 18 Kerne verfügbar. Doch warum kann man nicht einfach beliebig viele, zusätzliche Prozessorkerne auf den Sockel packen?

## Einschränkungen

Egal ob SMT, SMP oder eine Mischung aus beidem – ein grundsätzliches Problem bleibt bestehen. Bei grafisch intensiven Anwendungen wie 3D-Spielen kann es zum Beispiel vorkommen, dass die Performance eines Computers sinkt, je mehr Prozessoren man hinzufügt.

Einen Grund haben wir schon bei SMP kennengelernt: Stillstände des Prozessors können immer dann vorkommen, wenn die externen Ressourcen überbeansprucht werden – denn die sind ja nicht beliebig skalierbar. Zwar lassen sich theoretisch auch Grafikkarten (Quad-SLI) oder Festplatten (RAID) mehrfach parallel betreiben, doch auch das würde irgendwann nicht mehr weiterhelfen.

Denn der zweite und wichtigere Grund für eine „Kern-Barriere“ liegt an anderer Stelle: der Software.

Diese lässt sich zwar in Grenzen in einzelne, voneinander unabhängige Teile (Threads) aufteilen – aber eben nicht beliebig weit.

## Amdahl‘s und Gustafson‘s Gesetze

Programmcode ist nicht beliebig zerlegbar, und das aus einer ganzen Reihe von Gründen. Einer der wichtigsten ist, dass es bestimmte Vorgänge gibt, die sich nicht zerlegen lassen und immer nur einmal ausgelöst werden. Dazu gehört etwa die Speicherverwaltung eines Prozesses. Selbst wenn der nicht parallelisierbare Teil eines Programmes nur winzig ist, so sorgt er dennoch dafür, dass das Programm immer wieder zur Synchronisierung angehalten werden muss – ganz egal, wie viele Kerne oder Prozessoren im Rechner stecken. Diese Gesetzmäßigkeit wurde schon 1967 postuliert und heißt nach nach seinem geistigen Schöpfer „Amdahl‘s law“.

Mittlerweile sind die Datenstrukturen aber deutlich flexibler geworden und das Gesetz erfuhr einige Änderungen, die John L. Gustafson 1988 in Anlehnung an Gene Amdahls Regel formuliert hat. „Gustafson‘s law“ ist optimistischer und sieht selbst bei mehr als 100 Prozessoren bzw. Kernen noch Performance-Steigerungen, sofern der Code parallelisierbar ist – allerdings nur geringe (siehe Grafiken rechts).

## Code-Probleme und Spezialfälle

Beide Gesetze zeigen aber, dass eine lineare Steigerung zumindest im Desktop-Bereich relativ utopisch ist. Zehn Kerne werden bei normalen Anwendungen niemals die zehnfache Performance eines einzelnen Kerns bringen. Dazu gibt es – gerade bei Spielen – viel zu viele Abhängigkeiten innerhalb des Codes, ein weiterer Grund, warum sich Programme nicht beliebig klein teilen lassen. Spätestens dann, wenn eine zeitkritische Unterroutine aufgerufen wird, etwa die Ausgabe des nächsten Frames, müssen alle anderen performancerelevanten Engine-Teile wieder synchron sein.

Wirklich (annähernd) optimal zerlegbare Aufgaben werden dagegen ohnehin lieber auf der Grafikkarte durchgeführt – diese ist mit ihren tausenden Recheneinheiten deutlich besser für solche Berechnungen geeignet. Zu den sehr gut parallelisierbaren Aufgaben zählen beispielsweise viele KI-Probleme, etwa Deep-Learning-Algorithmen, welche die immer gleichen Codepfade auf Millionen und Milliarden von sich ändernden Daten anwenden müssen. So ließ etwa Google vor einigen Jahren die komplette Youtube-Datenbank mit knapp zehn Millionen Standbildern von mehr als 16.000 Prozessoren durchforsten und klassifizieren. Teilweise wurden dazu GPUs verwendet, mittlerweile hat der Suchgigant sogar eigene, massiv parallele KI-Chips entwickelt.

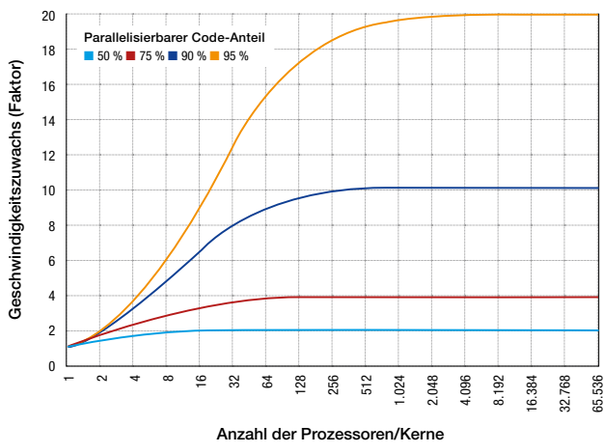
## Multicore-Designs

Es gibt eine ganze Reihe von Möglichkeiten, wie genau die CPU-Entwickler zwei oder mehr Kerne auf ein gemeinsames Substrat packen können. In der ersten und einfachsten Variante „klebten“ Intel-Ingenieure einfach zwei Kerne auf ein gemeinsames Substrat – fertig war der Doppelkern-Prozessor. Obwohl durch dieses Design viel Leistung verschenkt wurde, erreichten solche Prozessoren dennoch neue Bestmarken in Benchmarks, zumindest jenen, die eine Aufteilung in Threads vorsahen. Schon bald aber ging man zu „vollwertigen“ Designs über. Zwei oder mehr Kerne wurden von Anfang an als gemeinsame Einheit konzipiert und später auf die Wafer geätzt.

# Warum zu viele Kerne die Leistung bremsen können

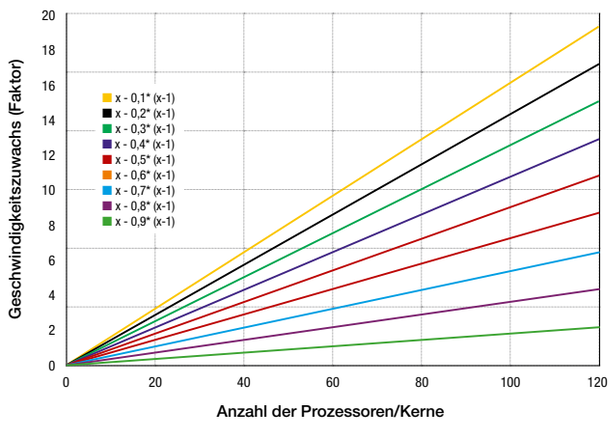
**Ist der Programmcode an einer Stelle nicht parallelisierbar, helfen auch Dutzende von Prozessoren kaum weiter. Selbst bei 95-prozentiger Parallelisierbarkeit erreicht man mit acht Kernen nur einen Geschwindigkeitszuwachs von rund 500 Prozent – jedenfalls in Amdahls Darstellung (unten). Aktuelle CPUs mit vier bis acht Kernen scheinen hier am Scheitelpunkt angekommen zu sein, denn auch die Kurve gut parallelisierbaren Programmcodes flacht oberhalb von vier Kernen spürbar ab. Diese Gesetzmäßigkeit formulierte Computerarchitekt Gene Amdahl bereits 1967 – sie ist heute aktueller denn je und als „Amdahl‘s Law“ bekannt.**

**Theoretisch mögliche Gewinne durch zusätzliche Prozessoren**




**Ende der 1980er-Jahre nahm sich der zwischenzeitlich auch bei AMD beschäftigte IT-Wissenschaftler John L. Gustafson die Regel noch einmal vor. Vereinfacht gesagt, geht Gustafson immer nur von einzelnen Aufgaben aus, etwa der Darstellung eines einzelnen Frames. Beschränkt man die Aussage auf nur diese eine Aufgabe, so könnten tatsächlich auch mehrere Dutzend Prozessoren noch einen Gewinn bringen – im Falle eines Spiele-Frames beispielsweise, wenn eine enorm hohe Polygonzahl vorhanden ist.**

**Theoretisch mögliche Gewinne durch zusätzliche Prozessoren, gemessen an einer einzelnen Aufgabe**



# Ihr Lieblings-Magazin auch digital lesen!

**PC Games Hardware gibt es in allen Variationen.**

**Sie müssen sich nur noch entscheiden, wie Sie Ihr Lieblings-Magazin lesen wollen: in gedruckter oder digitaler Form, einzeln oder im Abo.**

**Einfach, bequem, flexibel und immer dabei.**

**Oder wollen Sie einmal ein anderes Magazin ausprobieren?**

Alle Magazine gibt es für Tablet, Smartphone, Kindle Fire, im Browser und klassisch am Kiosk.

**shop.computec.de**

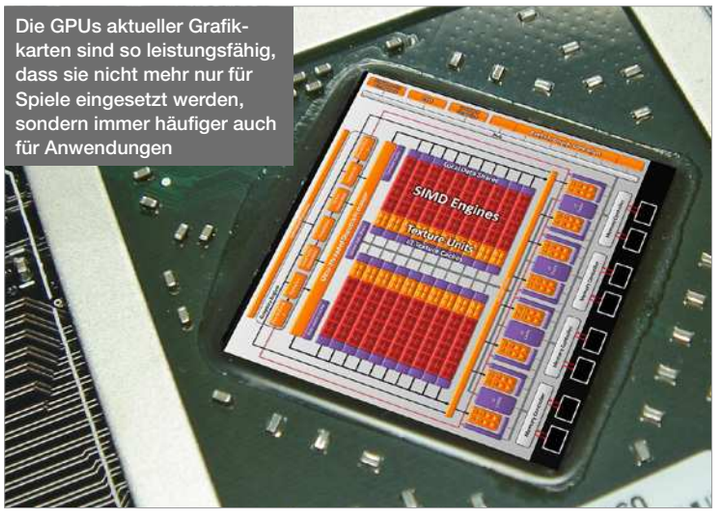
Oder einfach in Ihrem Store nach dem Namen des Magazins suchen.

Available on the App Store · JETZT BEI Google Play · Available on kindle fire

Ein Angebot der COMPUTEC MARQUARD MEDIA GROUP

# Grafikkarten

# So funktioniert eine Grafikkarte




Die GPUs aktueller Grafikkarten sind so leistungsfähig, dass sie nicht mehr nur für Spiele eingesetzt werden, sondern immer häufiger auch für Anwendungen

Aktuelle High-End-Grafikkarten sind vor allem auf schnellen Speicher angewiesen; üblich sind bis 16 GiByte GDDR5-Speicher mit bis zu 7 GHz Taktfrequenz

Vor mehr als 20 Jahren begann die Ära der 3D-Beschleuniger, einfacher Zusatzkarten, welche die Grafik der damaligen Spiele auf eine neue Ebene hievten.

## Erste GPU

Nach den ersten, eher experimentellen 3D-Beschleunigern Mitte der 1990er-Jahre (einzig der Voodoo-1-Chip von 3dfx war voll spieletauglich) erschien mit Nvidias Geforce im Herbst 1999 die erste „Graphics Processing Unit“ (GPU), ein Grafikchip, der nicht nur texturieren, sondern auch Polygone transformieren und beleuchten konnte – die CPU hatte somit eine Aufgabe weniger zu bewältigen. Mittlerweile ist die Komplexität wesentlich gestiegen. Doch wie funktioniert dieses komplizierte Stück Technik und was ist die Aufgabe von Jedi-Geheimeinheiten wie den ROPs oder der Polymorph-Engine?

## Grafikkarten – viel drauf

Die zweifellos wichtigste Komponente einer Grafikkarte ist der Grafikprozessor, auch als Grafikchip oder kurz GPU, analog zur CPU, bezeichnet. Dieser übernimmt sämtliche Berechnungen, die für die Grafikausgabe notwendig sind.

Unterstützt wird er von diversen anderen Bauteilen wie der Platine selbst, den Spannungswandlern samt Stromzufuhr und natürlich dem Grafik- oder auch Videospeicher (V-RAM). Für eine verständliche Erklärung der Grafikchip-Funktionen müssen wir auch darüber zunächst einige Worte verlieren.

## Die Platine der Grafikkarte

Das sogenannte Printed Circuit Board („PCB“), also die Platine, in der die Leiterbahnen eingelassen sind, stellt das Mainboard der Grafikkarte dar. Hierüber findet der komplette Datenverkehr ebenso statt wie die Erzeugung und Regulierung der benötigten Spannungen für den Grafikchip und seinen Speicher, die Ansteuerung der Monitoranschlüsse sowie die Verbindung mehrerer Grafikchips für SLI- oder Crossfire-Systeme.

Je schneller und damit meist auch stromhungriger eine Grafikkarte ist, desto komplizierter und damit auch teurer fällt das PCB der Karte aus. Die bis zu 4.096 Datenleitungen, mit denen der Speicher an den Grafikprozessor angebunden ist, sind in mehreren Lagen innerhalb des PCB untergebracht, sodass High-End-Modelle mitunter 12- oder gar 14-schichtige Platinen benötigen, was wiederum die Kosten bei der Produktion in die Höhe treibt.

## Unter Spannung

Ähnliches wie für das Board selbst gilt auch für die Spannungsversorgung bzw. die VRMs, was im Englischen für „Voltage Regulator Modules“ steht. Die VRMs wandeln die Zufuhr aus den 3,3- und 12-Volt-Leitungen des Netzteils in die nötigen Spannungen um und werden aus dem PCI-Express-Steckplatz selbst gespeist, über den 3,3 und 12 Volt mit insgesamt maximal 75 Watt (inkl. Toleranz) laufen.

Zusätzlich gibt es direkte Anschlüsse an das Netzteil. Die sechsadrige Variante liefert dabei ebenfalls 75 Watt; befinden sich acht Kabel am Steckerschuh, sind bis zu 150 weitere Watt möglich. Die Anzahl der Stromanschlüsse lässt auf den ungefähren (!) Stromhunger der Grafikkarte schließen:

- Bis 75 Watt (kein Extrakabel)
- 75 bis 150 Watt (1× 6-Pol)
- 150 bis 225 Watt (2× 6- oder 1× 8-Pol)
- 225 bis 300 Watt (1× 6- + 1× 8-Pol)

Je leistungsfähiger der Grafikchip, desto wichtiger ist eine gute, das heißt möglichst störungsfreie Versorgung mit der von ihm benötigten Spannung. Diese liegt für aktuelle Grafikchips in der Regel zwischen 0,7und 1,1 Volt und muss sehr schnell hin- und hergeschaltet werden, wenn der Grafikchip zwischen den verschiedenen Stromsparmodi hin- und herwechselt.

Auch die hohen Stromstärken von zum Teil über 100 Ampere müssen in Sekundenbruchteilen an die jeweils herrschenden Lastzustände angepasst werden können und dabei weite Bereiche überbrücken. Nicht selten sind achtphasig ausgelegte VRMs nötig, spezielle Übertakterkarten für Overclocking-Fans setzen auf eine noch höhere Anzahl an (Doppel-)Phasen. Das treibt direkt und indirekt über zusätzlich nötige Platinenbreite die Kosten in die Höhe.

## Anschluss gesucht

Über den heute üblichen PCI-Express-Anschluss gelangen die Daten zur Grafikkarte. Der üblicherweise mit 16 parallelen Leitungen verbundene Grafikkartensteckplatz (auch als PEG, „PCI-Express for Graphics“, bezeichnet) kann gleichzeitig Daten senden und empfangen und pro Richtung aktuell bis zu 16 GByte übertragen (PCI-E 4.0).

Stehen weniger parallele Bahnen (engl. „Lanes“) zur Verfügung, sinkt die Transferrate entsprechend – allerdings reagieren nur die wenigsten Spiele mit Fps-Verlusten auf einen Betrieb mit beispielsweise acht PCI-E-Lanes.

## RAM

Im Grafikspeicher werden (nach Möglichkeit) sämtliche Daten zwischengelagert, die zur Anzeige des Bildes nötig sind. Zwar befinden sich schnelle Zwischenspeicher (Caches) direkt im Grafikprozessor, oft genug jedoch muss auf Daten aus dem Hauptspeicher der Grafikkarte zurückgegriffen werden.

Für diese Fälle wird der RAM besonders bei High-End-Karten so schnell wie möglich ausgelegt und auch über viele parallele Datenleitungen mit dem Chip verbunden. Das wiederum kostet Platz und damit Geld – wie auch die Speicherchips selbst. Aktuell sind folgende Speicherstandards mit den angegebenen Taktraten gebräuchlich:

- HBM(2) (bis ca. 1.000 MHz)
- GDDR5X (bis ca. 5.700 MHz)
- GDDR5 (bis ca. 4.500 MHz)
- GDDR6 (bis ca. 7.750 MHz)

GDDR4 kam zwischenzeitlich auf einigen Radeon-Karten zum Einsatz (HD-3800-Reihe aus dem Jahr 2007), konnte sich aber nicht durchsetzen, da GDDR3-RAM den Taktnachteil schnell eingeholt hatte. Alte Grafikkarten nutzen oft DDR2- oder (G)DDR3-Speicher – kommt ein solches Modell in die engere Auswahl, achten Sie auf GDDR5-Bestückung.

## Grafikchip im Detail

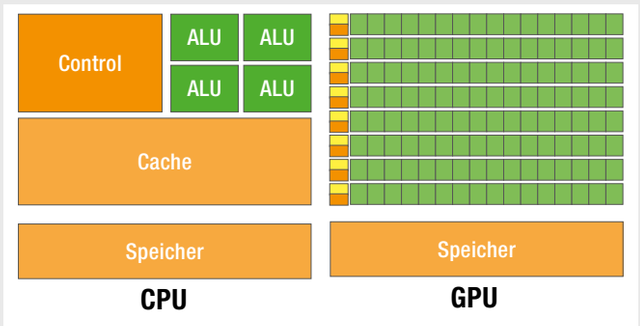
Auf den folgenden Doppelseiten stellen wir exemplarisch die Funktionsweise eines aktuellen Grafikchips vor und beschreiben anschaulich, welche Bedeutung die jeweiligen Baugruppen für die einzelnen Aufgaben in Spielen haben. Auf den dann folgenden Seiten erläutern wir alle wichtigen Funktionseinheiten noch detaillierter.

## Worin sich Grafikchip und Prozessor unterscheiden

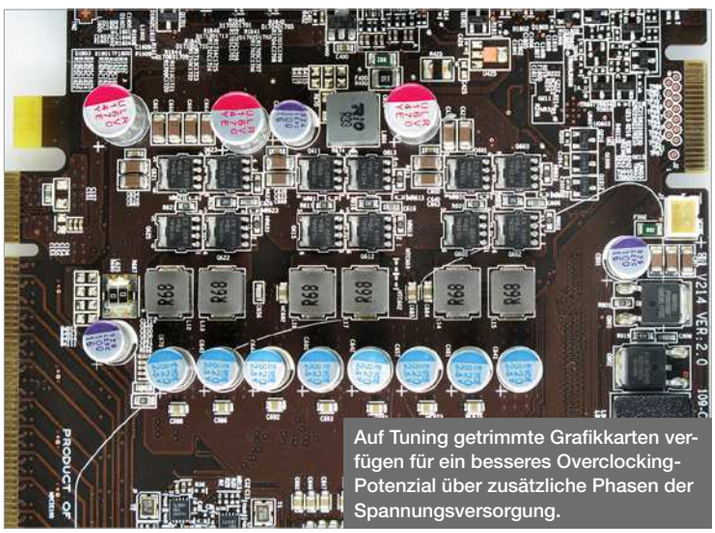
**Prozessoren werden grundsätzlich mit dem Ziel entwickelt, die unterschiedlichsten Aufgaben möglichst schnell auszuführen. Die unzähligen Erweiterungen des ursprünglichen x86-Befehlssatzes sind dafür ebenso ein Instrument wie die enorm hohe Taktfrequenz von vier Gigahertz und mehr.**

**Grafikkarten dagegen arbeiten vergleichsweise langsam – der Takt des Chips beträgt „nur“ rund 1,2 bis 1,9 Gigahertz und die Einsatzgebiete sind eng gesteckt. Dafür können mehrere hundert Rechenoperationen pro Takt durchgeführt werden, während es bei Prozessoren gerade mal eine Handvoll ist.**

**Grafikchips („GPUs“) und Prozessoren („CPUs“) werden manchmal auch mit Flugzeugen verglichen. Dabei sollen Prozessoren schnellen Düsenjets entsprechen, die einen oder zwei Reisende blitzschnell an einen anderen Ort bringen können. GPUs sind dagegen vergleichbar mit Passagierflugzeugen, die deutlich länger für die gleiche Strecke brauchen, dafür aber sehr viel mehr Passagiere gleichzeitig transportieren können.**




Ausführliche Darstellung auf Seite XX

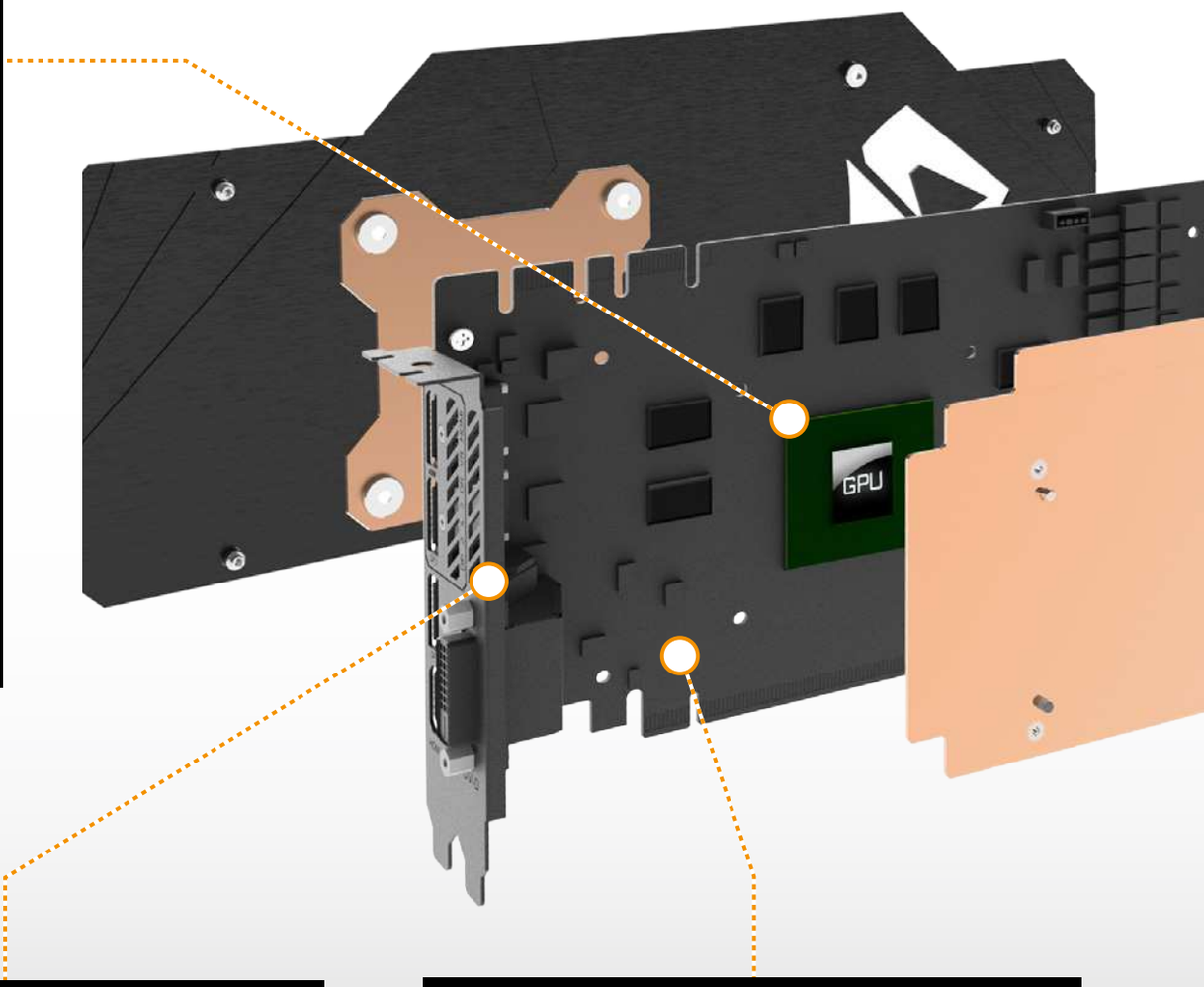


Auf Tuning getrimmte Grafikkarten verfügen für ein besseres Overclocking-Potenzial über zusätzliche Phasen der Spannungsversorgung.

# Die Grafikkarten-Bauteile im Detail

## Grafikchip

Der Grafikprozessor ist das Herzstück einer jeden Grafikkarte. Ohne ihn fände keine einzige Berechnung statt, das Bild bliebe aus. Moderne Grafikchips entstammen den Familien Turing (Nvidia) und Navi (AMD) und werden im sogenannten FinFET-Fertigungsprozess hergestellt. Der aktuelle Top-Chip für Spieler-Grafikkarten ist Nvidias TU102, welcher 18,6 Milliarden winzig kleiner Transistoren auf 471 Quadratmillimeter quetscht. Damit kann er nicht nur rund 510.000.000 Pixel pro Sekunde texturieren, sondern auch mit gut 42 Milliarden Dreiecken pro Sekunde hantieren. Für moderne Spiele und angepasste Applikationen ist jedoch die Rechenleistung der arithmetisch-logischen Einheiten, kurz ALUs, von besonderem Interesse. Hier stemmen moderne Grafikkarten dank maximal 4.608 Shader-ALUs mehr als 16 Tera-FLOPS (Billionen Gleitkommaberechnungen pro Sekunde) bei einfacher Genauigkeit. Zum Vergleich: Die 2006 revolutionäre Geforce 8800 GTX leistet nur 0,35 TFLOPS.




## Display-Ausgänge

Die mächtigste GPU ist – zumindest in einem Spiele-PC – nichts wert, wenn sie das Bild für sich behält. Damit Sie in den Genuss der berechneten Ergebnisse kommen, tragen moderne Grafikkarten mehrere Displayport-Ausgänge sowie mindestens einen, meist mehrere, HDMI-2.0-Ports, um dort Monitore anzuschließen. Der viele Jahre omnipräsente, hervorstehende DVI-Ausgang verschwindet langsam – die Referenzdesigns der RTX 2080 und RX 5700 verzichten beispielsweise längst darauf.

## Platine (PCB)

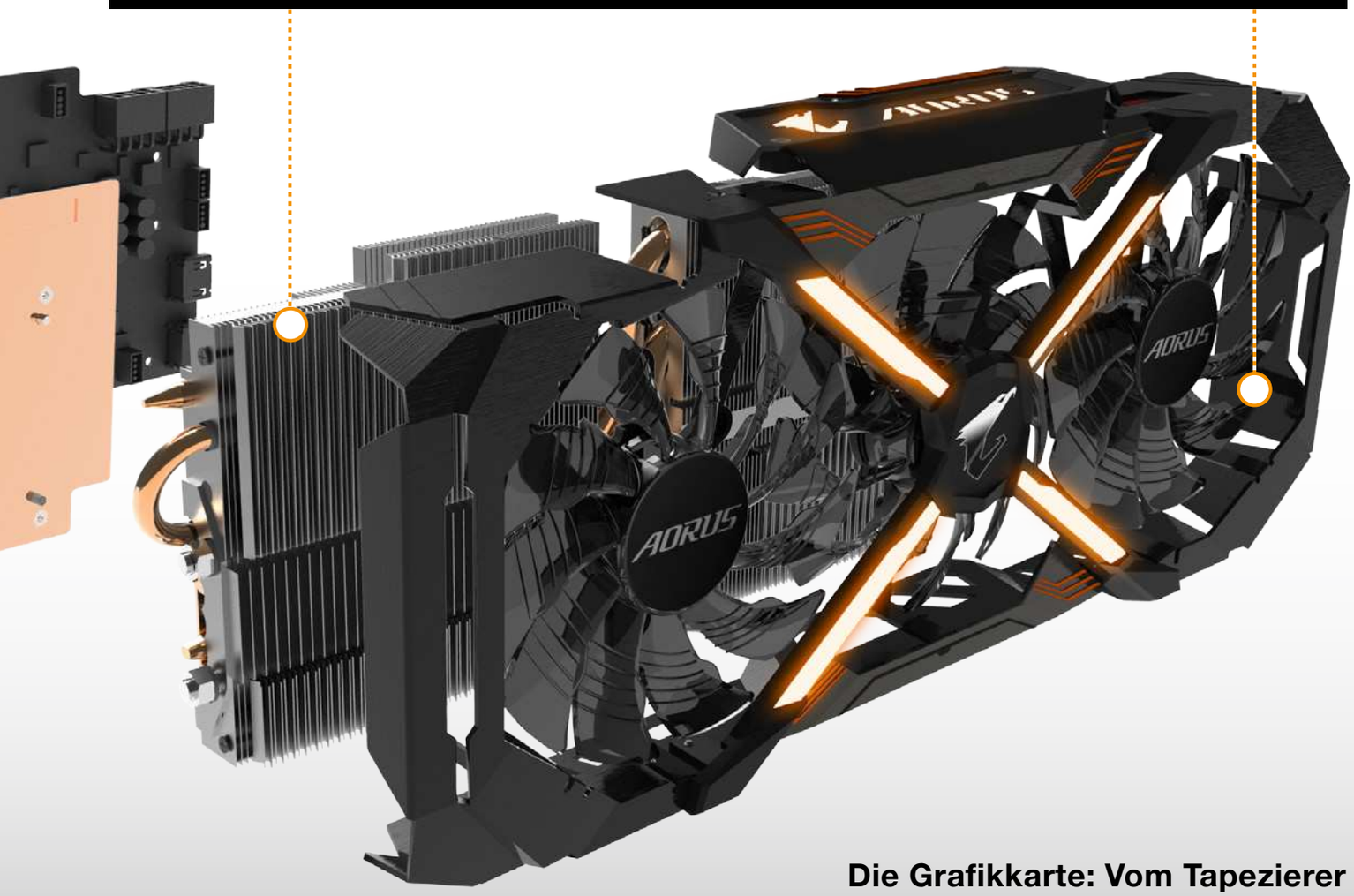
Das Printed Circuit Board, zu Deutsch etwa „bestückte Leiterplatte" und kurz PCB, ist das Mainboard der Grafikkarte. Auf ihm nehmen alle elektrisch notwendigen Bauteile Platz. Neben der GPU sind das unter anderem der Grafikspeicher (aktuell meist GDDR5/6), die für die Spannungsversorgung notwendigen Mosfets, Spulen und Kondensatoren, die Strombuchsen sowie zahlreiche weitere Bauteile (u. a. Widerstände), welche unter dem Begriff Surface Mounted Devices zusammengefasst werden.

## Kühlsystem

Im Laufe ihrer Existenz stieg die Grafikkarten-Leistungsaufnahme immer weiter, es schien kein Ende in Sicht. Zwischenzeitlich und begleitet von steigenden Strompreisen sowie wachsendem Umweltbewusstsein fand ein Umdenken statt. Moderne Mittelklasse-Modelle begnügen sich mit 200 Watt und die High-End-Modelle lassen sich auf höchstens 375 Watt treiben. Alles, was an Energie in die Grafikkarte fließt, wird in Wärme umgewandelt. Das heißt, jedes zusätzliche Watt ist mit Hitze gleichzusetzen, die es möglichst rasch abzuführen gilt. Moderne Kühler begegnen der Abwärme einer High-End-Grafikkarte mit reichlich Kühlmetall und ausgeklügelten „Sandwich"-Designs.

Wir haben mit der Gigabyte Aorus GTX 1080 Xtreme Edition bewusst ein Beispiel gewählt, das alle Register einer starken Luftkühlung zieht. Hierbei handelt es sich um eine 2,5-Slot-Konstruktion, welche den üblichen 3,5 Zentimeter breiten Dual-Slot-Rahmen (zwei Einbauplätze) sprengt, um die Kühlfläche zu erweitern. Doch der Reihe nach. Um die Wärmeableitung des Grafikchips und Speichers kümmert sich eine mehrere Millimeter dicke Kupferplatte („Base Plate"), welche wiederum Kontakt zu den Heatpipes des eigentlichen Kühlers aufnimmt. Bei Heatpipes handelt es sich um Kupferrohre, die mit einer Flüssigkeit gefüllt sind, welche nach Erwärmung verdampft, dadurch die Wärme transportiert, an anderer Stelle wieder erkaltet und das Spiel von vorne beginnen lässt. Moderne Heatpipes verfügen über einen Mindestdurchmesser von 6 mm, auf leistungsstarken (= heißen) Grafikkarten kommen jedoch auch Versionen mit 8 oder gar 10 mm Durchmesser zum Einsatz. Im Falle unseres Beispiels sind sechs Rohre installiert.

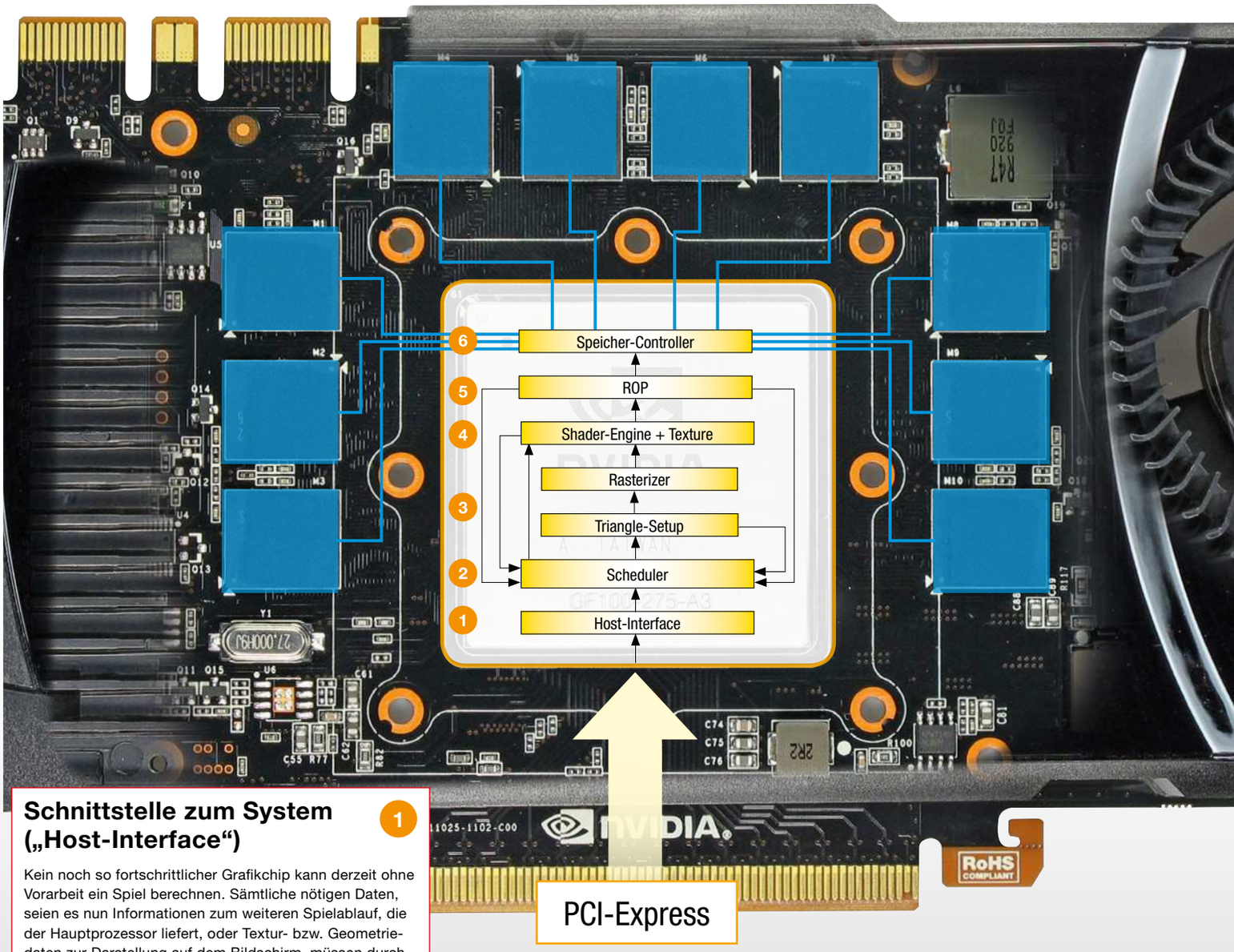
Die Heatpipes leiten die Wärme von den Hitzequellen weg an die zahlreichen Kühllamellen des Kühlerkorpus. Dieser besteht fast immer aus Aluminium und hat die Aufgabe, die Wärme an die Umgebungsluft abzugeben. Das funktioniert am schnellsten durch Luftbewegung, weshalb Gaming-Grafikkarten in der Regel über zwei, manchmal über drei, Lüfter verfügen. Gigabytes Stapeltechnik in unserem Beispiel erlaubt es, sogar drei Axiallüfter mit je 100 mm Durchmesser zu installieren – damit ist ein sehr hoher Luftdurchsatz schon bei geringen (= leisen) Drehzahlen möglich. Eine Rückplatte aus Kupfer komplettiert in unserem Beispiel den Wärmeaustausch.




## Die Grafikkarte: Vom Tapezierer zum Universalrechner

Anfang der 90er-Jahre des vergangenen Jahrtausends beherrschten Grafikkarten nur die Darstellung zweidimensionaler Bilder, bis Mitte der 90er die 3D-Revolution in Form von hübsch texturierter Polygongrafik begann. Heutzutage übernehmen Grafikkarten in Spielen unzählige Aufgaben und können auch Berechnungen abseits des Gamings respektive Grafik übernehmen – etwa Physik, Bildkonvertierung und vieles mehr.

# Wie eine Grafikkarte funktioniert




## Schnittstelle zum System („Host-Interface“) 1

Kein noch so fortschrittlicher Grafikchip kann derzeit ohne Vorarbeit ein Spiel berechnen. Sämtliche nötigen Daten, seien es nun Informationen zum weiteren Spielablauf, die der Hauptprozessor liefert, oder Textur- bzw. Geometriedaten zur Darstellung auf dem Bildschirm, müssen durch die Schnittstelle zum Wirtsystem, englisch auch „Host“ genannt. Über diese Schnittstelle kommt auch der vom Grafikkartentreiber in Maschinensprache übersetzte Kommandostrom, über den der Grafikkarte mitgeteilt wird, was als Nächstes zu berechnen ist.

Darin enthalten sind zum Beispiel auch Shader-Programme für die Pixelbeleuchtung oder Animationsberechnung. Zurzeit findet die Kommunikation zwischen Grafikchip und dem restlichen PC über die PCI-Express-3.0-Schnittstelle statt. Bei voller Bestückung des Steckplatzes mit 16 Datenbahnen sind darüber ca. 16 GByte pro Sekunde und Übertragungsrichtung möglich. Zurzeit stellt das Host-Interface keinen nennenswerten Flaschenhals dar, solange das Datenaufkommen den Grafikspeicher nicht überlastet.

## Der Vorarbeiter und Organisator („Scheduler“) 2

Das Steuerwerk des Chips. Hier werden die anstehenden Aufgaben verwaltet und den einzelnen Ausführungseinheiten wie Shader-Kernen oder Textureinheiten zugeteilt. Da es bis zum fertigen Pixel meist Dutzende, wenn nicht gar Hunderte von Arbeitsschritten sind, muss sich der Scheduler auch darum kümmern, den Fortschritt der Aufgaben zu protokollieren und teilweise abgearbeitete Befehlsketten neu in die Pipeline einzuspeisen. Das kann zum Beispiel passieren, wenn im Rahmen eines Direct-X-11-Programms über Geometryshader oder Tessellation zusätzliche Dreiecke erzeugt worden sind, die danach im Vertexshader-Programm animiert werden müssen und schließlich von einem Pixelshader mit der endgültigen Farbe versehen werden. Je nach Hardware fasst der Scheduler die pixelbasierten Aufgaben zu sogenannten Threads zusammen – Gruppen von Instruktionen, die auf die Shader-, Textur- oder Rastereinheiten aufgeteilt werden. In anderen Fällen bekommt er diese Gruppierung bereits vom Treiber aufbereitet serviert.

## Shadereinheiten #1: 4.1

Seit der mit der Geforce 8800 GTX/S eingeläuteten Direct-X-10-Generation verfügen alle Grafikchips über einheitliche Shader-Einheiten. Diese können, im Gegensatz zu DX9- oder älteren Grafikkarten, jede Art von Shaderprogramm ausführen – unabhängig von Geometrie-, Vertex- oder Pixelberechnungen. Der Vorteil liegt auf der Hand: Der Chip ist flexibler, da immer alle Einheiten beschäftigt werden können; ob die Spielszene nun sehr dreiecks- oder sehr pixellastig ist, spielt keine Rolle.

Im Rahmen der 3D-Berechnungen müssen die Pixel in spe jetzt allerdings Runden drehen und die Shadereinheiten mehrfach durchlaufen. Zum Beispiel …

- für Geometryshader und Tessellation
- für den Vertexshader
- für den Pixelshader

## Shader-Engine #2: Geometrie 4.2

Bevor die Pixel Pixel werden, haben sie schon einiges hinter sich. In einem ersten Schritt werden Vertices eingelesen und durch die Shadereinheiten im Rahmen der Vertex- und bei Direct X 11 der Hullshaderstufe berechnet – dabei werden die Koordinaten der Vertices in das System der 3D-Welt umgewandelt. Außerdem wird, falls nötig, der Tessellationsfaktor bestimmt, also der Grad der Unterteilung in feinere Geometrie. Diese neuen Vertices kommen dann im Falle von Direct-X-11-Tessellation zu der vorgesehenen Hardware und werden dort dem zuvor festgelegten Wert entsprechend verfeinert.

Als Nächstes werden die Domain- (bei DX11) und Geometryshaderstufe ausgeführt. In Ersterer wird die endgültige Position jedes Vertices berechnet, die auf den Werten des zuvor durchlaufenen Hullshaders basiert. In der Stufe des Geometryshaders wird dabei Geometrie hinzugefügt oder entfernt – je nach Vorgabe des Shaderprogramms. Danach folgt die sogenannte Viewport-Transformation und Perspektivenkorrektur zur Darstellung in der 3D-Ansicht. Sollen weitere Vertex-basierte Operationen ausgeführt werden, so wird das Vertex-Gitter in einen Zwischenspeicher geschrieben.

## Speichercontroller 6

Diese für die Leistung der Grafikkarte elementare Einheit ist meist in mehrere Blöcke unterteilt. Sie sorgen dafür, dass Textur- und Shadereinheiten möglichst schnell mit Daten versorgt werden.
Eine der wichtigsten Aufgaben der einzelnen Controller ist es, Speicherzugriffe so zusammenzufassen, dass möglichst wenig Verschnitt entsteht und die in der Regel aus 64 parallelen Datenleitungen bestehende Anbindung an die Speicherchips effizient ausgelastet wird.

## Setup und Rasterisierung 3

In dieser Stufe der Verarbeitung werden aus dem Vertexgitter erstmals echte Vielecke, indem die einzelnen Punkte untereinander verbunden werden. Das sogenannte Edge-Setup errechnet die Kanten der späteren Polygone. Hier setzt auch die erste Effizienzmaßnahme an: Durch Backface-Culling werden alle Dreiecke aus der Verarbeitung entfernt, deren Oberfläche vom Spieler weg zeigt und die deshalb nicht sichtbar wären. Der nächste Schritt ist die Rasterisierung. Darunter versteht man das Erzeugen von Pixeln entsprechend der Bildschirmauflösung. Der oder die Rasterizer rechnen aus den Kantengleichungen der Primitive nun die Entsprechung in Pixeln innerhalb der Bildschirmauflösung aus – ist Kantenglättung (Multisampling) aktiv, wird der Bedeckungsfaktor pro AA-Sample bestimmt, ansonsten einer pro Pixel. In der Regel arbeiten Rasterizer 16 Pixel pro Takt ab. Von hier aus geht es über die Z-Culling-Einheit weiter. Diese übernimmt auf Pixelebene, was das Backface-Culling auf Dreiecksebene geleistet hat: Die meisten Pixel, die im endgültigen Bild durch andere verdeckt und daher nicht sichtbar sind, werden von der weiteren Verarbeitung ausgeschlossen.

## Shader-Engine #3: Pixel 4.3

Endlich auf Ebene der Pixel angekommen, werden nun die Pixelshaderprogramme über die Shadereinheiten ausgeführt. Die dabei zum Einsatz kommenden Effekte sind vielfältig, zumeist aber mit irgendeiner Art von Texturen verknüpft. Um die Bereitstellung der Pixel-Tapeten kümmern sich die Textureinheiten, auch als TMU (vom englischen „Texture Mapping Unit“) bekannt, die bei heutigen Grafikchips meist als Quad-TMU direkt an die Shader-Engines angeschlossen sind. Bei Nvidias GP104-Chips teilte sich jeder Block von 16 Shader-Einheiten eine TMU. AMDs Cypress-, Juniper- und Redwood-Chips weisen ein 20:1-Verhältnis auf, die kleineren Cedar-GPUs der Radeon HD 5450 eines von 10:1. Polaris 10 hat ein Verhältnis von 16:1. Je mehr dieser Einheiten vorhanden sind und je effizienter sie arbeiten, desto weniger lang müssen die Shader auf die benötigten Werte warten, bevor sie weiterarbeiten können.

## Raster-Endstufen („ROPs“) 5

In die passend benannten Raster-Endstufen gelangen die Pixel bei heutigen Spielen erst kurz vor dem Ende ihrer Reise in Richtung Bildschirm. Die auch ROPs („Raster Operation Processor“) genannten Einheiten führen für die Antialiasing-Leistung und bei Schattenberechnungen wichtige Tiefentests durch und sind für das zum Beispiel bei Vegetationsdarstellung häufig genutzte Alpha-Blending zuständig.

# Wie 3D-Grafik in Computerspielen entsteht

Für Spiele, Filme und auch die Industrie ist dreidimensionale Computergrafik unverzichtbar. Doch wie funktioniert 3D-Grafik und warum ist es beispielsweise so schwer, runde Objekte abzubilden?

## Grafik aus dem Computer

Sie mag anfangs kompliziert erscheinen, doch 3D-Computergrafik zu verstehen oder selbst zu erschaffen, ist durchaus möglich. Während PC-Spieler vergleichsweise spät in den Genuss von 3D-Grafik kamen, gab es entsprechende Entwicklungen schon deutlich früher.

1972 entstand die kurze Filmsequenz A Computer Animated Hand, die von Edwin Catmull und Fred Parke erstellt wurde. Sie zeigt eine dreidimensionale Hand, die nicht nur als Drahtgittermodell (Wireframe), sondern mit schattierten Oberflächen (Polygonen) dargestellt wurde. Darüber hinaus war sie animiert, die Hand konnte sich also bewegen. Die beiden Schöpfer waren damals Studenten an der University of Utah. Catmull ist inzwischen Präsident der Walt Disney Animation Studios und war bereits bei der Gründung der Pixar Animation Studios ein Teil des Unternehmens. Parke veröffentlichte zwei Jahre nach A Computer Animated Hand noch eine Demonstration mit einem animierten, menschlichen Gesicht.

Auch vor den 1970ern gab es erste Gehversuche mit computergenerierten Grafiken, doch kamen aus Leistungsgründen damals vornehmlich Wireframe-Ansichten zum Einsatz. Es waren also ausschließlich die definierenden Kanten der Objekte sichtbar, nicht aber die Flächen. Ein Beispiel hierfür ist die 1963 von Edward E. Zajac veröffentlichte Animation eines Satelliten, der um die Erde kreist. Sie besteht aus einer sich drehenden Kugel und einem darum rotierenden Quader.

Arkadespiele der 1980er-Jahre (hier aus einem Battlezone-Derivat) setzten meist auf Vektorgrafik

## Passende Software kommt auf den Markt …

An Relevanz gewannen entsprechende Grafiken durch Programme, die die Erstellung von dreidimensionalen Objekten über eine Benutzeroberfläche ermöglichten. Zuvor mussten die Koordinaten von Eckpunkten von Hand eingegeben werden. In den 1980ern wurden mehrere Unternehmen gegründet, die bis heute bestehen und wichtige Programme aus diesem Bereich vertreiben: Autodesk startete 1982; das Unternehmen vertrieb zunächst Auto CAD (Computer Aided Design). 1990 erschien dann 3D Studio – der direkte Vorgänger von 3ds Max.

1983 wurde Alias Research, später Alias Systems Corporation, gegründet. Das Unternehmen entwickelte beispielsweise das Programm Poweranimator, 1998 kam dann die bekannte Software Maya auf den Markt, die inzwischen von Autodesk vertrieben wird.

## … und 3D-Spiele häufen sich

Während computergenerierte Grafiken für Animationen oder Spezialeffekte in Filmen schon früher zum Einsatz kamen, standen Spiele damals vor einem Leistungsproblem: Heimanwender hatten nur sehr begrenzte Rechenleistung und obendrein muss bei einem Spiel die Berechnung auch noch in Echtzeit erfolgen. Die grafische Darstellung wurde „live“ und damit interaktiv also erst später möglich.

Einer der ersten Titel mit gerenderten Flächen anstelle von Wireframes, 3D Monster Maze, erschien 1981 für den Sinclair ZX81. Dieser hatte einen Prozessor mit – aus heutiger Sicht lächerlichen – 3,25 Megahertz Takt verbaut. Das Spiel wurde in First-Person-Perspektive dargestellt; der Spieler musste darin einen Weg aus einem Labyrinth finden. Schon früher gab es Spiele mit Wireframe-Ansicht, etwa Spasim aus dem Jahr 1974. Ein verhältnismäßig flüssiges Spielerlebnis, auch in Wireframe-Ansicht, bot beispielsweise das von Atari entwickelte Battlezone vom November 1980. Zum Vergleich: Das berühmte Spiel Pac-Man erschien nur wenige Monate früher.

Ein Jahrzehnt später hatte sich die Grafik stark weiterentwickelt: Der Klassiker Wolfenstein 3D setzte auf einen Mix aus einer 3D-Spielwelt und 2D-Gegnern. Die Polygone waren hier bereits nicht mehr einfarbig, sondern mit Texturen versehen. 1995 kam der erste vollständig am Computer erstellte Spielfilm, Toy Story, ins Kino – und weitere Animationsfilme folgten. Die Grafik wurde nun langsam realistischer und vor allem massentauglich.

## Aufbau einer 3D-Szene

Da heutzutage fast alle Spiele 3D-Titel und entsprechende Technologien aus dem täglichen Leben keinesfalls wegzudenken sind, stellt sich die Frage nach dem Wie natürlich schnell. Insbesondere deswegen, weil in modernen Filmproduktionen grafische Niveaus erreicht werden, die man kaum mehr von der Realität unterscheiden kann.

Ob eine Computergrafik für ein Bild, einen Film oder ein Spiel erstellt wird, ist dabei auf Seiten der verwendeten Techniken kaum ein Unterschied. Der größte Unterschied liegt in dem Fokus auf eine hohe Leistungsfähigkeit bei Spielen – die Grundsätze bleiben aber im Prinzip identisch. Die folgenden Punkte sollen einen groben Überblick über den Aufbau einer dreidimensionalen Szene geben, wie sie zur Erzeugung computergenerierter Bilder verwendet wird. Als Szene wird dabei der virtuelle Raum bezeichnet, in dem sich alle Objekte mit ihren Einstellungen befinden.

## Geometrie

Soll auf dem Bildschirm später etwas Erkennbares ausgegeben werden, so muss der Computer zuerst wissen, wo sich Objekte befinden, wie diese angeordnet sind und wie sie aussehen. Objekte bzw. Modelle bestehen üblicherweise aus drei Elementen: Eckpunkten (Vertices), Verbindungskanten (Edges) und Flächen (Faces/Polygone). Diese sind voneinander abhängig: Eine Kante wird stets durch zwei Eckpunkte definiert; eine Fläche durch mindestens drei Kanten. Eckpunkte werden durch Koordinaten bestimmt, die die Position in der X-, Y- und Z-Achse angeben. Kanten und Flächen bauen auf den jeweiligen Eckpunkten auf.

Je nach Render-Engine können auch Kanten und Eckpunkte am Ende sichtbar sein, üblicherweise dreht sich die Geometrie aber um die Flächen – sie umschließen das Modell und bestimmen dessen Form. Dabei ist zu beachten, dass Flächen nicht, wie in der Realität, zwei Seiten haben, sondern nur eine. Betrachtet man am Computer die eine Seite einer Fläche, so ist sie wie zu erwarten sichtbar. Die Rückseite hingegen ist zumeist unsichtbar, weshalb man durch sogenanntes Backface Culling die nicht-sichtbaren Flächen weglässt, was Leistung spart.

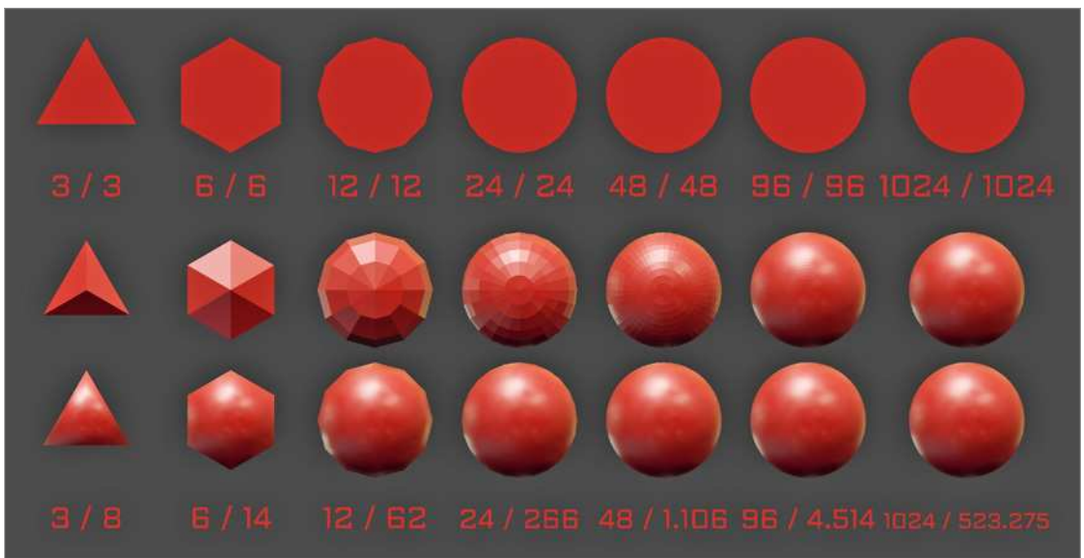
Am besten zu beobachten ist das, wenn man in einem Spiel in ein Objekt hinein gerät oder gar aus der Welt fällt. Dann kann man meist durch das Objekt beziehungsweise die Spielwelt hindurchsehen. Die entsprechende Ausrichtung einer Geraden wird als „Normale“ bezeichnet, einem auf der Fläche senkrecht stehenden Vektor, der eben in die eine oder in die andere Richtung zeigen kann.

Das alles mag sich ausgesprochen kompliziert anhören, ist aber in der Realität weit weniger problematisch: Entsprechende Programme bieten intuitive Wege, um Geometrie zu bearbeiten und zu erstellen. Und auch um die Grundprinzipien der Bearbeitung zu verstehen, ist nicht viel mehr notwendig als die heutzutage an Schulen unterrichtete Vektorrechnung.

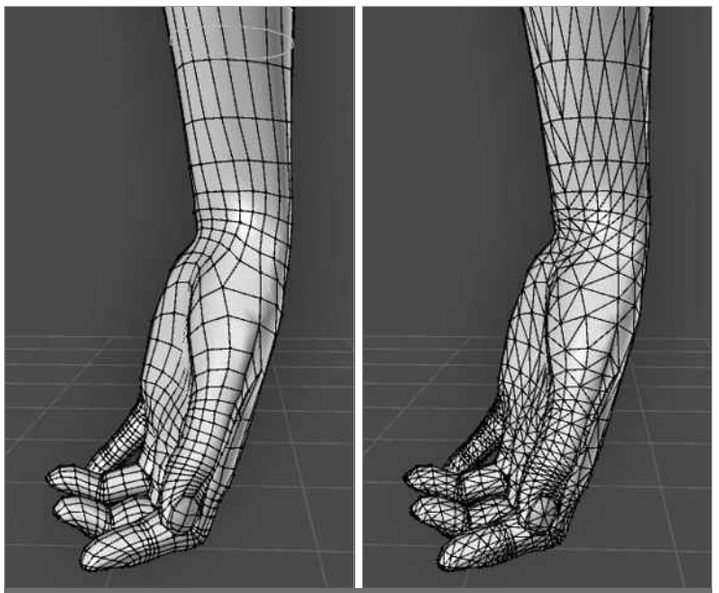
## Runde Objekte

Für einen 3D-Künstler ist all das Beschriebene einfacher Alltag, ein Problem ist vielmehr die saubere Erstellung von Geometrie. Das bedeutet, dass Objekte einfach zu bearbeiten sein sollen und vor allem nicht mehr Eckpunkte, Kanten und Flächen haben als notwendig – denn das kostet Speicher und Leistung. Ein Problem beim Einsparen von Geometrie gibt es vor allem bei der Darstellung runder Objekte: Flächen und Kanten sind immer gerade, weswegen Rundungen, um nicht kantig auszusehen, viel Geometrie verbrauchen.

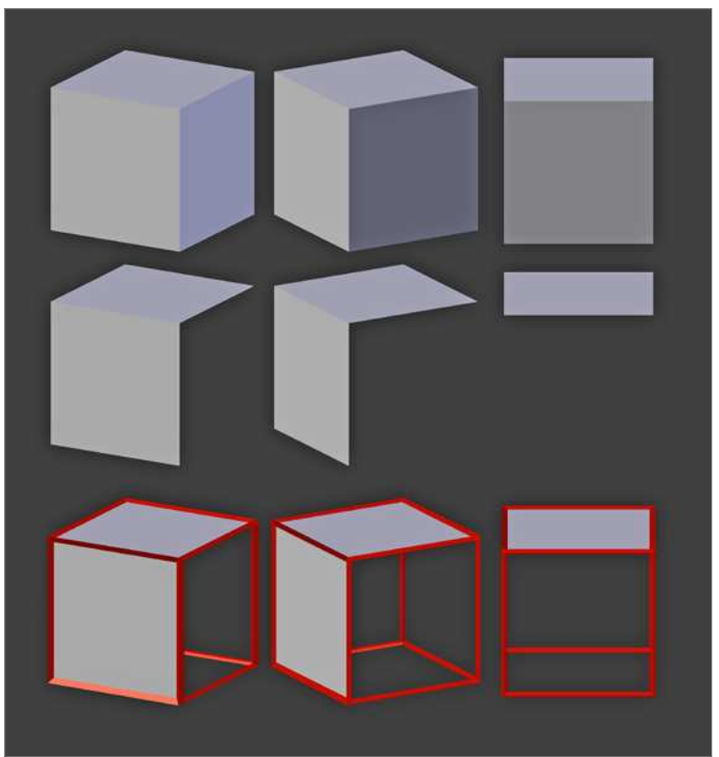
Render-Engines stellen Flächen intern als Dreiecke dar, denn egal welchen Punkt eines Dreiecks man verschiebt, die eingeschlosse-




Näherungen eines Kreises bzw. einer Kugel: In der ersten Zeile sind Kreise, in der zweiten Kugeln mit Flat Shading. In der dritten Zeile sehen Sie Kugeln mit Phong („Smooth“) Shading. Die Zahlen geben die Eckpunkte pro Kreis/des Gesamtobjekts an



Links: übersichtliches Modell aus Vierecken. Rechts: unübersichtliches Modell aus Dreiecken. Lila: Durch Kanten von Vierecken umschließt man einen logischen Bereich



Löscht man Seiten eines Würfels, sieht man, dass er hohl ist: 3D-Modelle setzen (meistens) nur auf Oberflächen. Dank „Backface Culling“ werden die Innenseiten nicht berechnet

ne Fläche ist immer „gerade“, also möglich. Verschiebt man jedoch bei einem Viereck einen der vier Punkte, so kann die Fläche „ungerade“ werden, also unmöglich; man kann dort nicht über alle vier Punkte eine gerade Fläche bilden. Die entsprechenden Programme zerteilen deswegen Mehrecke intern in Dreiecke, wofür es immer zwei Möglichkeiten gibt.

Obwohl am Ende in Dreiecken gerechnet wird, arbeitet man bei entsprechenden Modellen meistens mit Vierecken, um zwei gegenüberliegende Seiten zu erhalten. Das ist für den sogenannten Meshflow (auch Topologie genannt) wichtig: Man bildet mit den Vierecken Ringe um Teile des Modells. Das ist für einige interne Funktionen von Modellierungsprogrammen notwendig und sorgt zugleich für eine verbesserte Übersicht.

Sind alle benötigten Modelle erstellt, können sie im Raum angeordnet werden. Ausgehend von der in der Szene befindlichen Kamera kann die Render-Engine nun bestimmen, welche Objekte/Polygone weiter vorne liegen und welche verdeckt sind.

## Materialien und Texturen

Soll das Ergebnis auf dem Bildschirm jedoch nicht nur grau sein, sondern auch Farben enthalten, wird der Einsatz von Materialien (oft als „Shader“ bezeichnet; nicht zu verwechseln mit Shadereinheiten in GPUs) und Texturen notwendig. Erstere bestimmen die Eigenschaften der Oberfläche eines Objekts, Letztere wurden früher nur für die Farbgebung verwendet – heute jedoch für deutlich mehr.

Für Materialien gibt es verschiedene Möglichkeiten, um mit auftreffendem Licht zu reagieren. Sie können beispielsweise spiegelnd oder diffus sein – oder auch beides gleichzeitig, in einem beliebigen Verhältnis. Sehr alte Spiele haben keine Reflexionen, die Materialien sind also alle diffus.

Das Problem an spiegelnden Oberflächen ist, dass der Lichtstrahl abgelenkt wird und damit nach dem Auftreffen auf das erste Objekt noch weiterverfolgt werden muss; das kostet Leistung. Einfarbige Materialien, die höchstens mit dem umgebenden Licht interagieren, sind hier schonender, aber eben auch deutlich weniger ansehnlich. Mit moderner Hardware und damit mehr Leistungsreserven sind spiegelnde Oberfläche jedoch möglich geworden. Je nach Implementierung sind sie wirklich „realistisch“ oder nur die Realität nachahmend.

## Textur-Tricks

Render-Engines bieten für gewöhnlich fertige Materialien an und unterteilen diese beispielsweise in metallische, nicht-metallische, transparente etc. Damit Modelle nicht überall dieselbe Farbe haben, wird auf diese eine Textur gelegt. Jede Fläche erhält so sozusagen eine „Tapete“, die aus einem Bild entnommen wird.

Heutzutage werden Texturen jedoch nicht nur für die Farbe eines Objekts eingesetzt. Weitere Anwendungszwecke sind zum Beispiel Normal- und Displacement-Maps. Mit diesen können die Eigenschaften der Objektoberfläche weiter beeinflusst werden. So kann man zum Beispiel auf derselben Fläche mehrere Materialien darstellen: Ist die Textur an einer Stelle weiß, wird Material A verwendet, andernfalls Material B.

Die in den letzten Jahren massiv angestiegene Verwendung von nicht nur höher auflösenden, son-

dern auch mehr Texturen ist einer der Gründe für die immer größer werdenden Spiele. Die Farbtextur braucht beispielsweise drei Farbkanäle (Rot, Grün, Blau), ebenso wie eine Normal-Map. Displacement-Maps kommen mit einem Kanal (Schwarz/Weiß) zurecht.

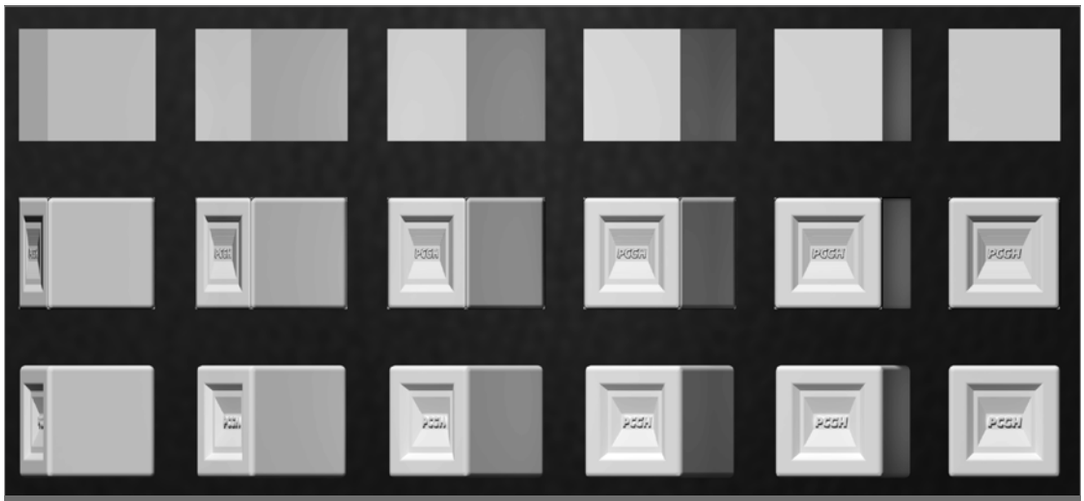
## Es werde Licht!

Nun befinden sich nicht mehr nur einfache Objekte in unserer Szene, sondern solche mit Materialien und Texturen, also Objekte mit Farben. Damit diese am Ende stimmig dargestellt werden, ist nur noch eines notwendig: Licht, das mit den Materialien interagiert. Dafür gibt es verschiedene Ansätze: Render-Engines verfügen meist über mehrere Arten von Lichtquellen, beispielsweise unsichtbare Punkte oder Geometrien, die Licht ausstrahlen. Auch wird oft die Umgebung verwendet: Die Himmelstextur beleuchtet die Szene individuell.

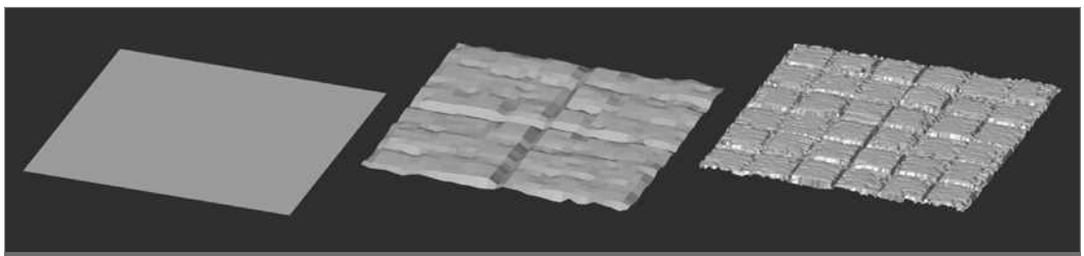
Lichtquellen werden dabei, wie im echten Leben auch, durch die Parameter Farbe, Helligkeit und Größe definiert. Ersterer verändert beim Auftreffen des Lichts auf ein Material die dargestellte Farbe, während die Helligkeit auch ebendiese im Endergebnis beeinflusst. Die Größe wiederum bezieht sich auf die entstehenden Schatten: Ist eine Lichtquelle unendlich klein, so sind alle entstehenden Schatten an den Karten hart. Es gibt also keinen Übergang vom schattierten zum nicht schattierten Bereich. Macht man die Lichtquelle größer, wird der Übergang weicher – zumindest beim Offline-Rendering.

## Dynamik kommt ins Spiel

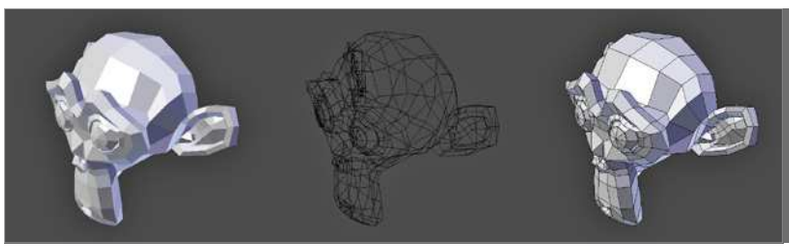
Mit den bisher beschriebenen Bestandteilen haben wir alles, um ein Bild zu berechnen. Der Vorteil in 3D-Welten liegt aber in der Interaktivität und genau diese gilt es



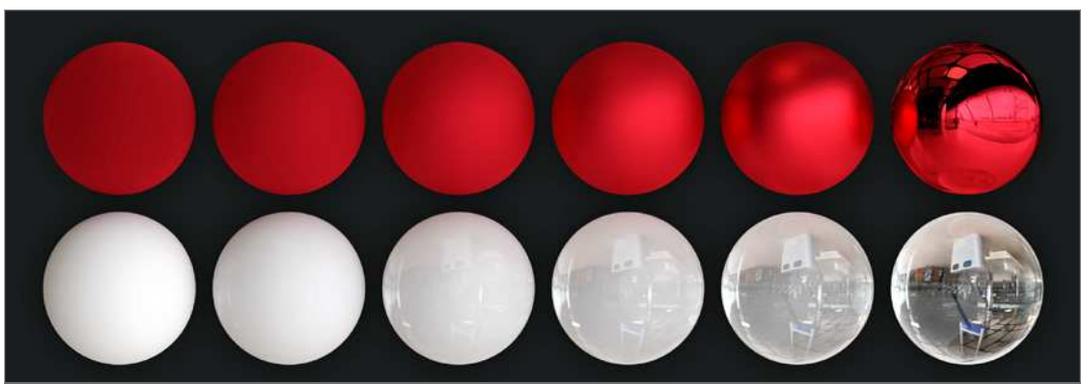
Mit Bump-Maps kann man Vertiefungen vortäuschen. Oben: Das Objekt ohne Bump-Map. Mitte: Das Objekt mit Bump-Map. Unten: Das Objekt mit modellierten Details (Vertiefungen durch echte Geometrie, nicht durch Texturen)



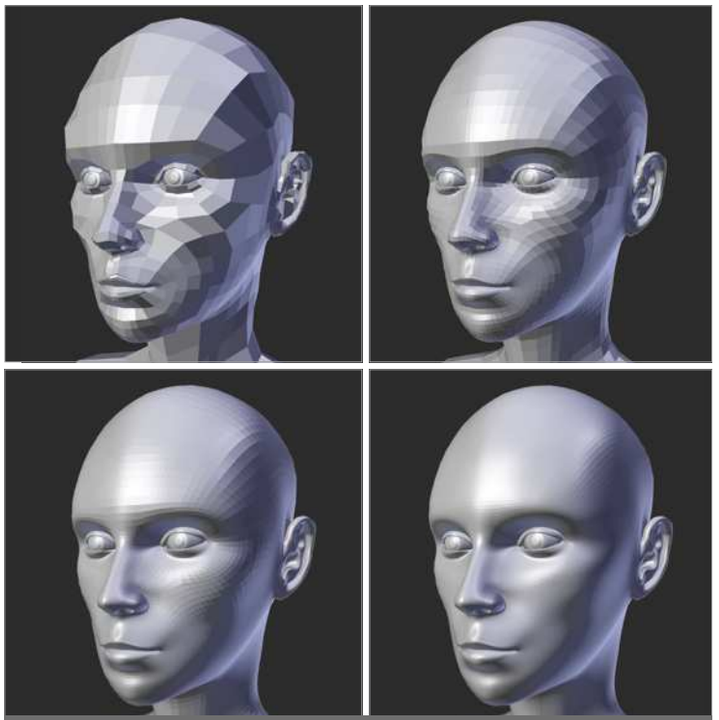
Durch Displacement-Maps können Objekte mit höherem Detailgrad dargestellt werden. Nähere Objekte werden automatisch weiter unterteilt (z. B. Catmull-Clark-Algorithmus/Tesselation) und anhand der Displacement-Map verschoben. Bei zu niedriger Mindestdistanz sieht man eine Formänderung, wenn man sich Objekten nähert



Links: Solid. Mitte: Wireframe. Rechts: Solid + Wireframe: Die Solid-Ansicht zeigt das Modell mit Flächen an, die Wireframe-Ansicht nur mit Kanten. Nur die Eckpunkte anzuzeigen ist unüblich, da man dann die Struktur des Objekts kaum mehr erkennt



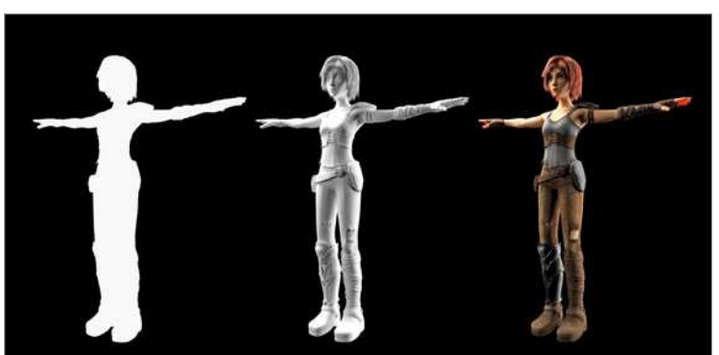
Materialien. Oben: Stufen zwischen diffus und glänzend. Unten: Stufen zwischen diffus und Glas. Während Raytracer echte Spiegelungen bieten, werden bei Spiel-Engines oft Tricks zur Realisierung eingesetzt



Der Catmull-Clark-Algorithmus erzeugt zusätzliche, abrundende Geometrie und funktioniert am besten mit Vierecken. Jede weitere Stufe vervierfacht die Polygon-Anzahl. Zu hohe Einstellungen können schnell zu Leistungs- und Speicherproblemen führen, deswegen wird ergänzend zumeist Phong („Smooth“) Shading eingesetzt

## Gängige Texturen/Maps

| Art der Textur/Map | Zweck |
|---|---|
| Diffuse/Albedo | Grundsätzliche Farbgestaltung des Objekts |
| Normal/Bump/Parallax | Beeinflusst Lichteinfall auf ein Objekt, kann Vertiefungen in geraden Ebenen vortäuschen |
| Displacement/Tessellation | Kann tessellierte (zerteilte) Flächen verschieben, um eine nahe Oberfläche mit zusätzlicher Geometrie zu berechnen |
| Specular | Bestimmt, wo ein Objekt stärker/schwächer glänzt |
| Transparency (manchmal in Diffuse/Albedo enthalten) | Kann ein Objekt durchsichtig machen, beispielsweise oft bei Gewächsen eingesetzt |



Modell: „Sintel“ von Andy Goralczyk (Quelle: cloud.blender.org). Links: Modell ohne Schatten/Beleuchtung. Mitte: Modell ohne Materialien („Clay Render“). Rechts: Modell mit Texturen, Materialien und Beleuchtung

noch zu erlangen. Die Koordinaten von Objekten können natürlich verschoben und rotiert werden. Das würde bereits reichen, um die Kamera in verschiedene Positionen zu bringen. Möchte man die Modelle jedoch dazu bringen, ihre Form zu verändern, so müssen diese zuerst „geriggt“ und animiert werden.

Das Rig ist ein Skelett, das in einen Körper eingesetzt wird. Möchte man einen Menschen riggen, so würde man zwischen allen relevanten Gelenken einen „Bone“ einsetzen. Anschließend wird jedem Vertex im Bezug auf jeden Bone ein „Gewicht“ (eine Stärke) zugewiesen. Bewegt man nun einen Bone, dann werden alle dazugehörigen Vertices, abhängig von ihrem Gewicht, mitbewegt. Würde man beispielsweise einen Unterschenkel bewegen, so wäre es unsinnig, den entsprechenden Knochen mit Eckpunkten am Ohr zu verknüpfen.

Neben Knochen werden häufig noch sogenannte Shape Keys eingesetzt. Ein Shape Key macht in der Praxis nichts anderes, als ausgewählte Eckpunkte in eine angegebene Position zu verschieben. Diese Funktion wird oft verwendet, um durch das Rig entstehende Fehler zu kaschieren oder zum Beispiel Muskelbewegungen hinzuzufügen.

### Von der Theorie in die Praxis

Bei all der Theorie stellt sich natürlich die Frage, wie man die genannten Dinge in die Praxis umsetzt. Die Zeiten, in denen Koordinaten und Verbindungen von Hand in Tabellen eingetragen wurden, sind glücklicherweise schon lange vorbei. Für alle genannten Arbeitsschritte gibt es eigenständige Programme und auch Komplettlösungen in sehr großer Anzahl. Meistens ist Software zur Modellierung auch in der Lage, die Modelle entsprechend zu texturieren, die Materialien und Lichter einzustellen, Modelle zu riggen und das Ergebnis am Ende zu berechnen. Trotzdem werden auch bei Komplettlösungen oft noch andere Programme eingesetzt, die auf einen einzelnen Arbeitsschritt spezialisiert und dort dementsprechend überlegen sind.

### Kostenlose 3D-Programme

Bekannte Programme der Industrie sind in etwa 3ds Max und Maya, beide aus dem Hause Autodesk. Sie bieten beispielsweise die beschriebenen Funktionen. Die von professionellen Studios häufig eingesetzten Programme sind jedoch oft zu teuer für Privatanwender, weswegen hier kostenlose (Open-Source-) Alternativen attraktiv erscheinen – zumindest, wenn man zuerst einmal in den Bereich der 3D-Kunst hinein schnuppern möchte. Das wohl mächtigste unter den kostenlosen Programmen ist Blender.

Blender startete ursprünglich auch als kommerzielle Software, doch mit dem Bankrott des entwickelnden Unternehmens Not a Number Technologies sah Chefentwickler Ton Roosendaal die Möglichkeit, das Programm gegen 100.000 Euro, die per Crowdfunding gesammelt wurden, als Open-Source-Software zu veröffentlichen. Seit damals, 2002, wird Blender von der Non-Profit-Organisation Blender Foundation unter der Leitung von Roosendaal verwaltet und weiterentwickelt. Die Lizenz ist GPL v2, die Software ist dementsprechend vollständig kostenlos.

Das Programm wird von Einsteigern zu Beginn oft als kompliziert empfunden. Mit entsprechendem Lehrmaterial ist es aber durchaus auch für Hobbyisten möglich, die Verwendung dieser mächtigen Software zu erlernen.

# FETT!

Zwei DVDs!

PLUS 16 Seiten extra!

Inkl. Top-Vollversion!

**PC GAMES – WISSEN, WAS GESPIELT WIRD**

WWW.PCGAMES.DE | AUCH DIGITAL ERHÄLTLICH

Ein Produkt der COMPUTEC MARQUARD MEDIA GROUP

# Wie Rastergrafik und Raytracing funktionieren

Mit dem Slogan „Grafik neu erfunden“ bewirbt Nvidia seine Turing-Grafikkarten der RTX-Reihe und spielt damit auf das Kernmerkmal entsprechender Grafikprozessoren an: Raytracing

Bild: Remedy

Laut Remedy ist die Raytracing-Technologie im Forschungszustand und „weit davon entfernt, in einem Computerspiel eingesetzt zu werden“ – hier eine Demo, die Nvidia zeigte

Das Thema „Raytracing in Spielen“ taucht mit schöner Regelmäßigkeit alle paar Jahre wieder auf – zuletzt von Nvidia und Microsoft ins Gespräch gebracht. Wir erklären die Unterschiede zur klassischen Rasterisierung und zeigen den aktuellen Stand der Entwicklung auf.

## Raytracing-Versprechen

Ankündigungen von Nvidia zufolge soll Raytracing in Spiele bald zum Standard werden. „Einmal mehr“, mag man entgegnen, denn entsprechende Versuche gab es schon häufiger, unter anderem von Intel, wo man 2009 mit „Larrabee“ den Grafik- und Videospielemarkt revolutionieren wollte. Das Projekt wurde sang- und klanglos eingestellt. Raytracing bei Intel erlebte jedoch vor wenigen Monaten eine überraschende Wiederauferstehung: Intel möchte bis 2020 einen entsprechenden Grafikprozessor auf den Markt bringen. Ist also doch mehr dran am Thema Raytracing?

## RTX und DXR

Zurück zu Nvidia: Der Chiphersteller glaubt, dass jetzt die Zeit gekommen ist, in der Grafikkarten stark genug sind, um Raytracing massentauglich zu machen. Zusammen mit Microsoft stellt der Chiphersteller „RTX“ vor, das als Back-End zu Microsofts neuer Direct-X-Raytracing-API, kurz „DXR“, dient. Schon in den vergangenen Jahren ging es auf Nvidias GTC-Hausmessen immer wieder um Raytracing, allerdings vor allem im Zusammenhang mit der Filmproduktion. Dort wird die Render-Technik schon lange für Spezialeffekte beziehungsweise ganze Animationsfilme eingesetzt, da nahezu beliebig viel Zeit zum Rendern vorhanden ist, um das optisch schönste Resultat zu erzielen.

RTX nennt Nvidia seine Features, um die Berechnung von Raytracing zu beschleunigen, und die Bündelung von Interfaces. RTX ist Teil von Microsofts DXR, das Spieleentwickler nutzen können, um das Rendern über Raytracing in Direct-X-12-Titeln zu ermöglichen. Nvidia möchte ein Gameworks-Modul anbieten, das DXR auf einem „High-Level“ offenlegt.

Zuerst soll die Technik die bisher genutzte Rasterisierung ergänzen, später vielleicht ganz ersetzen. Zu Beginn soll die Technik in den Bereichen eingesetzt werden, in denen sie den größten Effekt hat: Bei der Berechnung von Schatten und Spiegelungen. Mit namhaften Studios wie DICE (Battlefield, Star Wars Battlefront) und 4A Games (Metro) sowie Engine-Entwicklern wie Epic Games (Unreal-Engine) und Unity Technologies (Unity-Engine) gab und gibt es bereits wichtige Interessenten und Entwicklungspartner für die Technik.

## AMD

Auch AMD arbeitet nach Angaben von CEO Lisa Su an Raytracing-GPUs. Doch unabhängig davon hat AMD auf der Fachmesse Game Developers Conference (GDC) eine hauseigene Open-Source-Raytracing-Engine vorgestellt, die exakt den gegenteiligen Ansatz von Microsoft und Nvidia verfolgt: Statt einen Rasterizer um einen Raytracer zu ergänzen, um eine höhere Qualität zu erhalten, soll bei „Radeon ProRender“ ein Raytracer Teile mit Rasterisierung auf Vulkan-Basis rendern, um Zeit zu sparen. Außerdem soll AMDs „Radeon Rays Raytracer“ in Unity integriert werden – allerdings nur, um das Erstellen von sta-

tischen Beleuchtungsmodellen zu beschleunigen.

## Raytracing-Anwendungen

Abseits von Spielen kämpfen AMD und Nvidia auch bei Raytracern und Software für den produktiven Einsatz um Unterstützung und Marktanteile. Aus diesem Grund entwickeln die beiden Unternehmen auch eigene Lösungen: Bei AMD beispielsweise das erwähnte „Radeon ProRender“, bei Nvidia „Iray“.

Der parallele Einsatz von Raytracing und Rasterisierung ist für diese Geschäftsfelder ebenso interessant wie für Spiele. Auch wenn per Raytracing gerenderte Bilder oder Filme nicht in Echtzeit präsentiert werden müssen, ist eine möglichst schnelle und leistungssparende Berechnung trotzdem von großem Interesse, um Zeit und Kosten zu sparen.

Für Spieler relevanter, wenn auch indirekt, ist der Einsatz derartiger Methoden in Programmen für Spieleentwickler – wie jene des bereits genannten Entwicklers Allegorithmic. Für Spiele werden oft zuerst hochauflösende Objekte erstellt, die anschließend auf leistungssparsamere Versionen heruntergerechnet werden. Das Erstellen von den komplexeren Modellen ist für die darstellenden Programme eine Herausforderung: Schnell kommt es zu Framedrops und Ladezeiten im Viewport, was die maximale Qualität und somit auch das Endergebnis limitiert.

Zudem ermöglicht eine flüssigere Darstellung auch ein schnelleres Arbeiten – man hat mehr Zeit, die man dem Objekt und damit dem Endergebnis widmen kann. Die Kombination der beiden Techniken kann also Spielen auch dann helfen, wenn sie gar nicht direkt eingesetzt wird.

## Geschwindigkeit vs. Realismus

Um die Vorteile der Kombination beider Techniken zu verstehen, muss man sie getrennt betrachten; und auch die Marketingaussagen der entsprechenden Unternehmen mitunter etwas dämpfen. Stark vereinfacht kann man sich die beiden Ansätze so vorstellen: Rasterisierung ist ein extrem schneller, aber unrealistischer Weg, ein Bild zu berechnen, während Raytracing in der Theorie zwar „fotorealistisch“ arbeitet, dafür aber die Berechnungszeit gegen unendlich tendiert.

## Rasterisierung

Rasterisierung ist die bei Spielen dominierende Technik und aktuelle Grafikkarten sind für diese Art der Berechnung sehr gut geeignet. Der Algorithmus betrachtet zu Beginn eines jeden Frames die in der Szene enthaltenen Polygone. Entscheidend für das Rendering ist dabei als erstes die Frage, welche Polygone überhaupt zu sehen sind. Ist in einer Szene ein Objekt zu sehen, so werden in jedem Fall die von der Kamera abgewandten Polygone gleich zu Beginn verworfen (Backface Culling). Dasselbe passiert, wenn ein Polygon des Objekts vollständig außerhalb des Sichtfelds der Kamera liegt (Viewport Culling), zum Beispiel zu weit oben.

Alle verworfenen Polygone sind für den nachfolgenden Rendering-Prozess nicht mehr relevant, sie werden nicht mehr betrachtet. Das kann, wie gerade beschrieben, einzelne Polygone betreffen, aber auch ganze Objekte, die von weiter vorne liegenden Objekten verdeckt werden. Auch alles, was hinter der Kamera liegt, wird nicht beachtet. Die übrigen Polygone werden sinnbildlich direkt vor die Kamera gesetzt, sodass sie den gesamten Bildausschnitt verdecken. Nun kann die Kamera das Bild Pixel für Pixel ab-

Bild: Futuremark

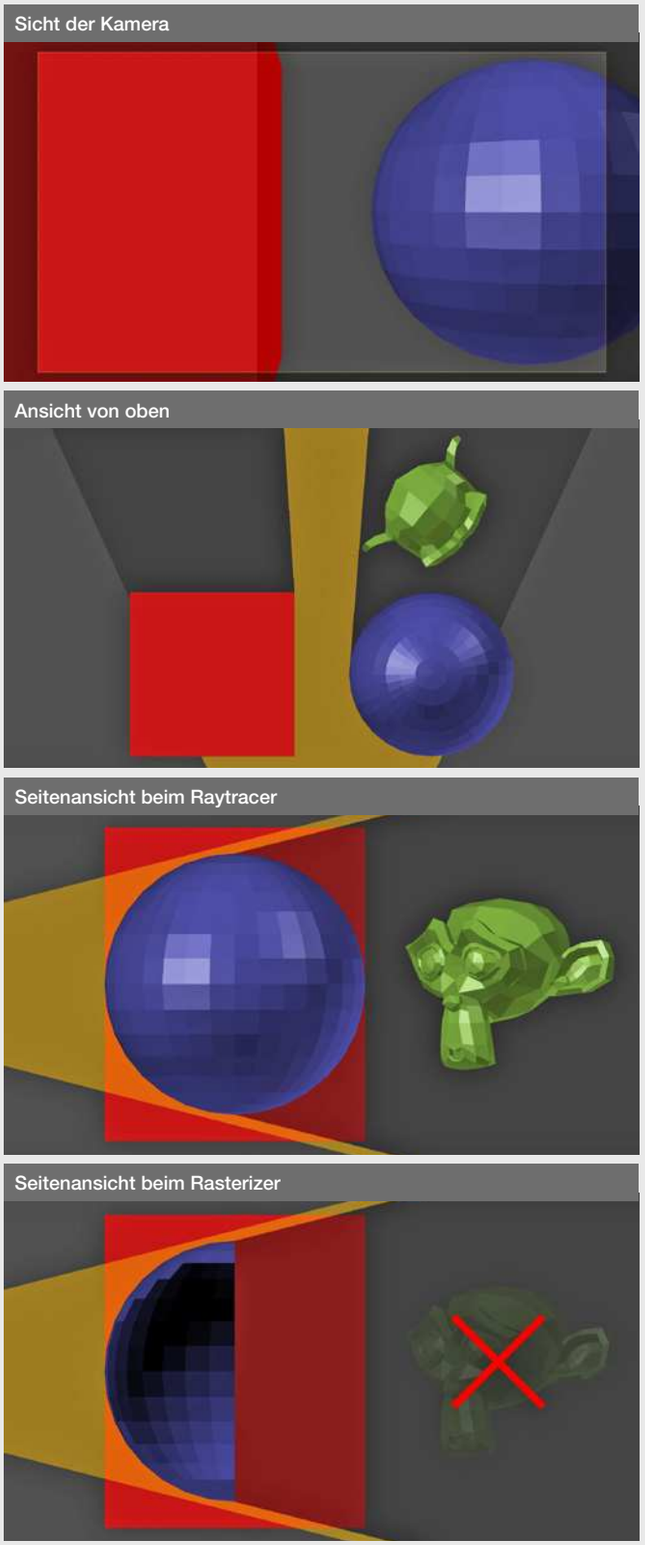
Die Futuremark-Demo aus dem 3DMark zeigt Direct-X-Raytracing im Einsatz. Oben kommt eine niedrig auflösende, vorberechnete Spiegelung zum Einsatz; unten Raytracing

Bild: Remedy

Remedy (Max Payne, Quantum Break) ist mit seiner Northlight-Engine einer der Technologiepartner von Nvidia und Microsoft für die Direct-X-Raytracing-API. In Kürze soll das RTX-Spiel „Control“ auf den Markt kommen

## Geometrie in 3D-Spielen

**Die Unterschiede bei der Geometrie: Raytracer betrachten die gesamte Szene, Rasterizer verwerfen Flächen außerhalb des direkten Sichtfelds.**




tasten und für jedes einzelne Pixel entscheiden, welches Polygon ihn bestimmt. Da die Texturen direkt auf den Polygonen aufliegen, ist eine Einfärbung nach dem entsprechenden Bereich auf der Textur des Gesamtobjekts kein Problem.

### Kantenflimmern

Ist ein Polygon nicht perfekt entlang der X-/Y-Achse ausgerichtet, sondern hat eine schräge Kante, dann entsteht hier der gut bekannte Treppeneffekt, der ohne Anti-Aliasing auftritt. Der einfachste Ansatz zur Behebung dieses Problems ist es, schlicht eine höhere Auflösung zu berechnen und diese anschließend herunterzuskalieren – das liefert eine sauberere Kantenführung und weniger Flimmern.

### Lichtquellen

Möchte man das Bild jedoch in, nach aktuellen Standards, akzeptabler Grafik haben, dann müssen durch die Rasterisierung noch viele zusätzliche Dinge berechnet werden. Der gerade beschriebene Weg liefert zum Beispiel keinerlei Information über Beleuchtung und Schattierung der Polygone. Dafür wird die ausgedünnte Szene noch einmal aus Sicht der Lichtquellen berechnet und anschließend, in Kombination mit der Sicht der Kamera, entschieden, welche Pixel beleuchtet sind und welche nicht.

Da die Lichtquellen bei dieser Methode unendlich klein sein müssen, da man sonst nicht eindeutig aus Sicht der Lichtquelle in die Szene sehen könnte, gibt diese Methode keine Information über die Härte der Schatten. Die Kanten sind also immer scharf gezeichnet, weiche Übergänge müssten wieder durch zusätzliche Berechnungen realisiert werden.

Nun fehlen noch viele weitere Details, wie die zahlreichen Varianten von Reflexionen, globaler Beleuchtung und so weiter. All diese werden, wie die Lichtberechnung, mit Tricks und weiteren Rechenschritten hinzugefügt. Am Ende besteht ein solches Bild also aus einer Vielzahl an Rechenschritten.

### Raytracing

Im Gegensatz dazu hat Raytracing einen etwas leichter verständlichen, da physikalischen Ansatz. Das Verfahren kann sowohl auf Prozessoren als auch auf Grafikkarten ausgeführt werden und kommt beispielsweise bei Produktvisualisierungen und in Filmen zum Einsatz.

Raytracing betrachtet die komplette Szene, das heißt alle Objekte mit allen Polygonen und allen Lichtquellen zugleich. Von der Kamera aus werden Lichtstrahlen in die Szene geschickt, die an den Objekten abprallen. Die Strahlen prallen so lange ab, bis sie auf eine der in der Szene verteilten Lichtquellen treffen. Üblicherweise gibt es aber einen Maximalwert für die Anzahl der Reflexionen, damit sich Lichtstrahlen nicht an einem Punkt „verfangen“ können. Hat der Strahl nach einer bestimmten Anzahl von Berechnungen noch keine Lichtquelle entdeckt, dann wird er verworfen. Zusätzlich versucht der Lichtstrahl nach einer bestimmten Menge der Reflexionen, sich direkt mit der Lichtquelle zu verbinden. Der Kollisionspunkt muss also nur im Sichtbereich einer Quelle liegen und nicht durch Zufall die exakten Koordinaten treffen.

Den Weg, den der Lichtstrahl zurückgelegt hat, merkt sich der Raytracer – denn erst, wenn er die Lichtquelle gefunden hat, kann die Farbe bestimmt werden. Die resultierende Farbe hängt schließlich vom Licht ab, das heißt der gesamte Weg muss am Ende zurückverfolgt werden. Nun kann immer der

aktuelle Wert des Lichtstrahls berechnet werden. Am Anfang strahlt die Lichtquelle direkt auf das erste Material. Danach ist der Lichtstrahl bereits anders: Das erste Material hat ihn verändert. Außerdem ist die Intensität des Lichts nun weniger stark. Das Licht kann theoretisch auch von der Lichtquelle aus zur Kamera gesendet werden, das Prinzip ist dabei identisch.

## Realistische Lichtverhältnisse

Durch die Simulation von Lichtstrahlen liefern Raytracer ein verhältnismäßig realistisches Bild, wobei einige Effekte, die durch Rasterisierung schwer darzustellen sind, durch Raytracing sehr leicht erzielt werden. Globale Beleuchtung, also das gegenseitige Beeinflussen von Objekten und Licht, ist bei einem Raytracer kein Problem. Auch Spiegelungen und sanfte Schatten sind bereits von Haus aus mit dabei. Das Realisieren von volumetrischen Effekten wie Subsurface Scattering (Streuung von Licht im Körper, beispielsweise bei Wachs und Haut) sind bei einem Raytracer ebenso gut zu erreichen.

Das Problem an der Rechenweise von Raytracern ist, dass ihnen ein einzelner Lichtstrahl pro Pixel nicht ausreicht: Es wird eine sehr hohe Anzahl von Lichtstrahlen benötigt, um ein sauberes Bild zu erzeugen. Ansonsten kommt es zu Bildrauschen. Die Ergebnisse der Strahlen werden pro Pixel aufaddiert. Das, was am Ende zu sehen ist, ist also ein Durchschnittswert aller einzelnen Ergebnisse (Akkumulation).

## Hohe Rechenzeit

Da sich das Rauschen nur mit jeder Verdopplung der gesendeten Lichtstrahlen halbiert, kann die benötigte Rechenzeit sehr schnell ansteigen. Bilder, die pro Pixel nur einen einzigen Lichtstrahl gesendet hätten, wären vermutlich mit etwas Optimierung durchaus flüssig darstellbar; nur sähe das Endergebnis nicht annähernd erwartungsgemäß aus.

Zudem ist auch der grundlegende Ressourcenverbrauch hoch: Im Gegensatz zur Rasterisierung werden hier keine Daten verworfen, man betrachtet also die ganze Szene gleichzeitig. Positiv ist wieder-

## Bildrauschen bei Raytracing

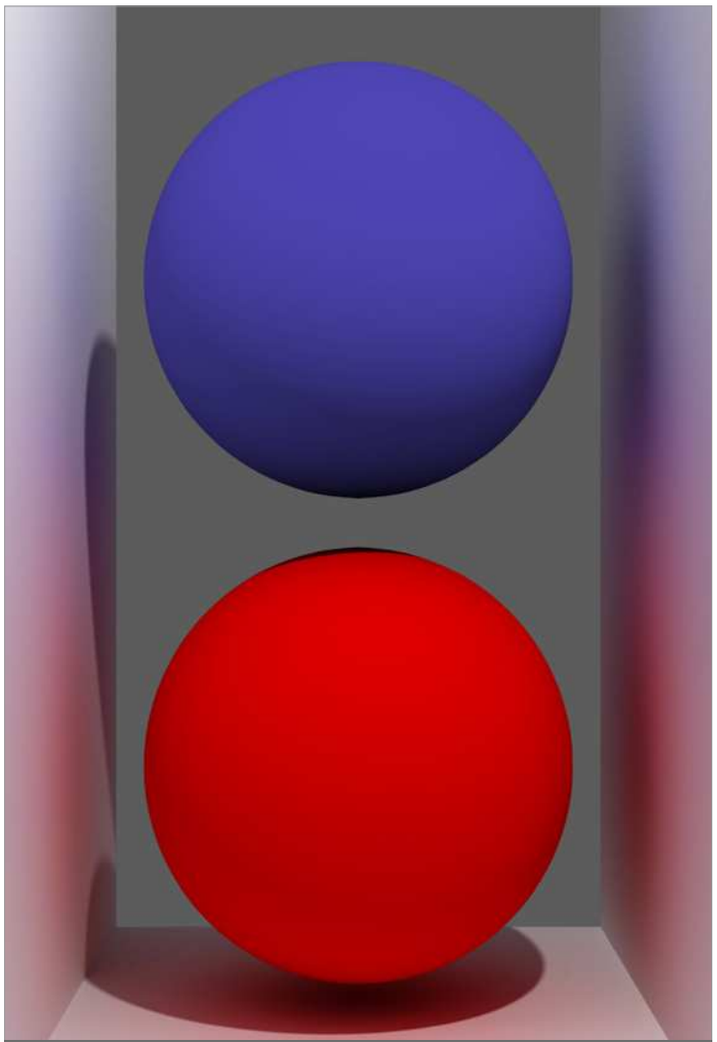
**Per Raytracing berechnetes Bild in verschiedenen Qualitätsstufen: In dieser Szene wird der vom Schirm geworfene Schatten erst mit vielen Lichtstrahlen deutlich sichtbar, vorher wird er vom Rauschen überlagert.**

1 Sample (Anzahl der Lichtstrahlen)

2 Samples (Anzahl der Lichtstrahlen)

8 Samples (Anzahl der Lichtstrahlen)

32 Samples (Anzahl der Lichtstrahlen)



Mit Raytracing ist globale Beleuchtung leicht umzusetzen. Beleuchtete Objekte beeinflussen sich gegenseitig, so färben die beiden Kugeln leicht auf die Wand ab

Rasterizer können grundsätzlich nur harte Schatten werfen (links) und müssen auf „Tricks“ zurückgreifen. Bei Raytracern (siehe Mitte und rechts) sind hingegen auch weiche Übergänge im Schattenwurf möglich

um, dass sich die Berechnung der Lichtstrahlen sehr gut parallelisieren lässt. Das ist auch der Grund, warum AMDs Bulldozer-CPU-Architektur oft mit Raytracing-Demos präsentiert wurde – hier konnten die vier Module schon damals voll ausgelastet werden.

## Beide sind nicht perfekt

Bei den GDC-Ankündigungen von Microsoft und Nvidia konnte man den Eindruck erhalten, dass Rasterisierung „schlecht“ und Raytracing „gut“ sei. In der Praxis sind beide aber nur Grundkonzepte der Berechnung, die hier schematisch beschrieben wurden. Ein Raytracer könnte jeden Lichtstrahl auch nur ein einziges Mal reflektieren lassen und dann, wenn er die Lichtquelle nicht direkt trifft, sofort verwerfen. Die Berechnung wäre so schneller gelöst, aber sichtbare Spiegelungen gäbe es keine und das Resultat sähe schlechter aus als man es von heutigen Spielen gewohnt ist. Im Gegenzug könnte man einer Spiel-Engine auch mehr Zeit als die üblichen 33,33 Millisekunden (30 FPS) spendieren und damit zusätzliche Effekte wie realistischere Spiegelungen simulieren lassen.

In der Praxis werden beide Verfahren angepasst, um einen bestmöglichen Kompromiss zu finden; je nach Ansatz, aber auch je nach Einsatzgebiet in unterschiedlicher Ausprägung. Bei Rasterisierung sind beispielsweise Shadow und Reflection Maps nicht mehr wegzudenken, ohne sie gäbe es in Spielen weder Schatten noch Spiegelungen. Etwas fortgeschrittener sind die Implementierungen von globaler beziehungsweise indirekter Beleuchtung oder auch Screen Space Reflections.

## Bildrauschen

Bei Raytracern wiederum ist man bereit, etwas Realismus zu opfern, wenn sich dadurch die Rechenzeit deutlich verringert. Ein gutes Beispiel hierfür sind Denoiser (Noise = Rauschen; Denoise = Entrauschen), die am Ende des Renderings das verbleibende Rauschen entfernen. Bei manchen komplexen Szenen kann man auch nach Dutzenden Stunden der Berechnung noch ein leichtes Rauschen sehen, und da, wie bereits erwähnt, jede Halbierung des Rauschens eine Verdopplung der Lichtstrahlen erfordert, würde ein subjektiv rauschfreies Bild einen nicht tolerierbaren Zeitrahmen beanspruchen.

## Denoiser

Hier kommen Denoiser ins Spiel. Diese gibt es beispielsweise für Pixars Raytracer Renderman oder den kostenlosen Raytracer Cycles. Denoiser sind Algorithmen, die während des Renderings Informationen speichern, anhand derer das Rauschen bestmöglich aus dem Endergebnis entfernt wird – es sind also keine nachträglichen Filter, wie sie von Grafikbearbeitungsprogrammen eingesetzt werden.

Einen Denoiser setzt auch Nvidia für das Raytracing in Spielen ein, um das Endresultat trotz der geringen Berechnungszeit ansehnlich zu machen. Dieser soll auf neuronalen Netzen basieren und in Spielen eine akzeptable Optik liefern. In einer Demonstration wurde er von Nvidias Tensor-Kernen betrieben, in Spielen wird Denoising von Compute Shadern vollzogen. Im Gegensatz zu den bereits auf dem Markt befindlichen Denoisern von Raytracern hat Nvidia dabei den Vorteil, dass Spiele zumeist zusammenhängende Bilder produzieren. Das bedeutet, dass zur Eliminierung des Rauschens oder gar von komplett schwarzen Pixeln, die manchmal auftreten, unter Umständen auch Ergebnisse der vorherigen Bilder berücksichtigt werden können. Zudem wird für Bewegt-

bild fast nie ein rauschfreies Ergebnis vorausgesetzt. Filme weisen beispielsweise auch ein Rauschen auf.

## Was haben Spieler davon?

Wie bereits ausgeführt, hat Raytracing bereits von Haus aus Stärken, bei denen die Rasterisierung tricksen muss. Vor allem betrifft das die Berechnung von Spiegelungen und Schatten, weshalb diese Bereiche als erstes durch Raytracing abgelöst werden sollen.

Ganz neu sind Raytracing-ähnliche Ansätze in Spielen allerdings nicht: Die häufig verwendeten Screen Space Reflections funktionieren bereits vergleichbar. Die Engine berechnet, wo der Lichtstrahl nach der spiegelnden Oberfläche auftreffen würde, und nimmt diesen Farbwert an. Die Umsetzung hat aber einige Schwächen, die Raytracer nicht haben: Beispielsweise können nur Dinge gespiegelt werden, die bereits im Sichtfeld sind. Aus diesem Grund wird das Endergebnis oft noch automatisch nachbearbeitet.

Eine partielle Implementierung von „echtem“ Raytracing kann Spielen aber in der Tat dabei helfen, schönere Spiegelungen und feinere Schatten zu erhalten. Nicht vergessen darf man dabei, dass das noch lange nicht die aus Filmen bekannte und perfekte Grafik mit sich bringt. Raytracing kann zwar einige Dinge besonders gut darstellen, eben beispielsweise Reflexionen, aber es kann viele andere Probleme nicht lösen.

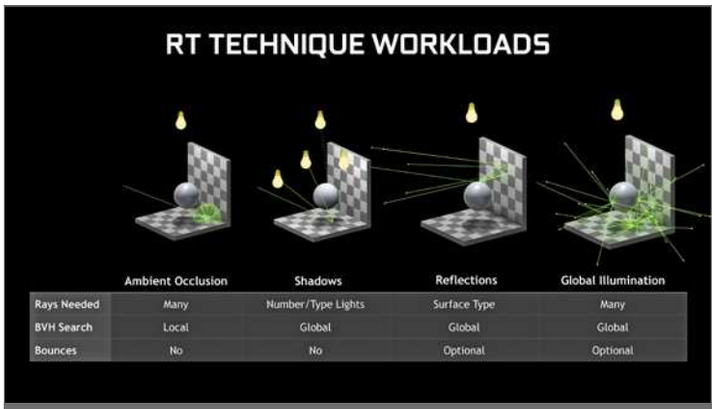
## Raytracing ist kein Wundermittel

Denkt man an Raytracing, dann ist der gedankliche Weg zu CGI-Szenen, wie perfekt realistischen Naturlandschaften, gar nicht mehr so weit. Es stimmt zwar, dass hierfür Raytracing eingesetzt wird, doch in Spielen wird man so etwas trotzdem noch länger nicht sehen. Die Art des Renderings nimmt nämlich keinen Einfluss darauf, was darunter liegt: Objekte und Texturen.

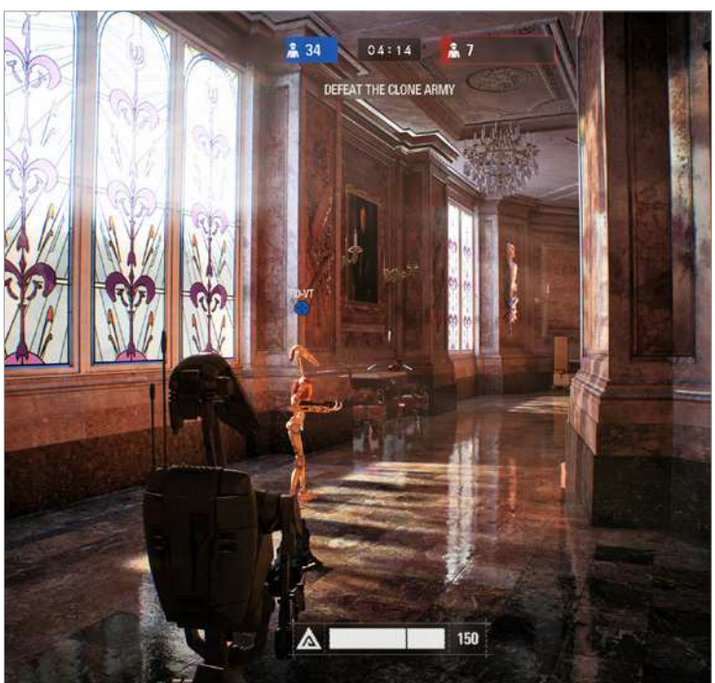
Eine Wiese mit Millionen von einzelnen Grashalmen, die alle mit verschiedenen, hochauflösenden Texturen versehen sind, ist für einen Raytracer kein Problem – so die Theorie. In der Praxis werden solche Bilder jedoch meistens auf Prozessoren berechnet, weil Grafikkarten schlicht nicht genügend Arbeitsspeicher dafür haben. Unter diesen Bedingungen kann das auch jeder Rasterizer darstellen. Die Art der Berechnung ist dafür nicht entscheidend. Zudem haben Raytracer auch keine zusätzlichen Texture-Maps, die sie verwenden könnten: Normal-, Specular- und Displacement-Maps gibt es sowohl im Raytracing als auch in der Rasterisierung.

Darüber hinaus gibt es auch physikalische Effekte, die nicht in jedem Raytracer umgesetzt werden. Beispielsweise spiegeln Oberflächen stärker, wenn der Einfallwinkel sehr steil ist. So kann eine lackierte Holzfläche in der Frontalansicht komplett matt erscheinen, betrachtet man sie von der Seite, spiegelt sie hingegen leicht. Für ein wirklich realistisches Bild müsste dieser Effekt beachtet werden, eine Voraussetzung für die Bezeichnung als Raytracer ist die Unterstützung aber nicht.

In erster Linie dürfte die Integration von Raytracing in Spielen also wirklich nur den Bereich Spiegelungen und die Schattenberechnung verändern. Immerhin sind mittlerweile viele RTX-Spiele angekündigt und man darf gespannt sein, wie weit Entwickler die Grafikhardware ausreizen können.




Diese vier Effekte werden derzeit mittels Raytracing in Spielen umgesetzt. Die Aufstellung stammt von Nvidia und staffelt die Effekte aufsteigend nach Anspruch




Auch wenn sie mit Tricks hinzugefügt werden müssen, können schöne Beleuchtungseffekte und Reflexionen auch bei Rasterizern erreicht werden. (Spiel: Star Wars Battlefront 2)

# Moderne Grafikeffekte: Reflexionen

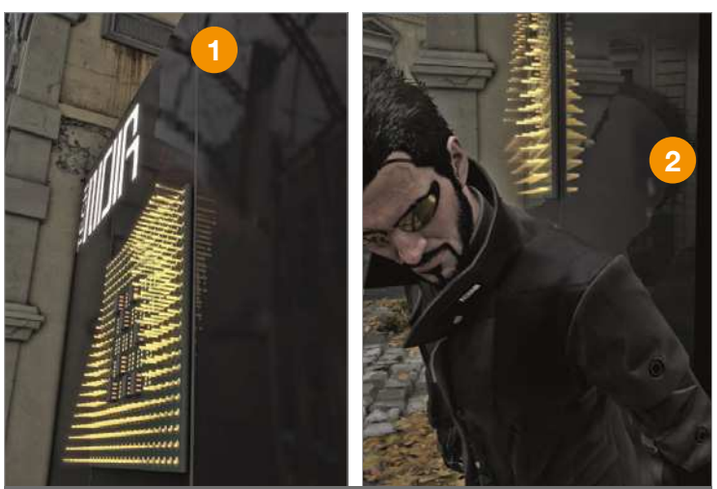
The Division zeigt insbesondere nachts einige sehr schicke Reflexionen und nur wenige Artefakte

Reflexionen tragen in Spielen viel zur Atmosphäre bei: Nass glänzender Asphalt, spiegelnde Pfützen, Wasseroberflächen oder funkelndes Metall – ohne reflektiertes Licht beziehungsweise dessen nährungsweiser Simulation würden Szenen in Spielen viel von ihrer Glaubwürdigkeit verlieren.

Zudem dienen Reflexionen als stilistisches Mittel, ein Film wie Blade Runner würde ohne sich auf dem nassen Asphalt widerspiegelnde Neonreklame einen großen Teil des charakteristischen Looks einbüßen.

Wollen die Entwickler in ihrem Spiel Reflexionen nutzen, so stehen sie indes vor einem Problem: Eine korrekte Spiegelung einer Szene, beispielsweise in einer Pfütze am Boden, würde aufgrund der Komplexität moderner Titel voraussetzen, dass die komplette Szene inklusive Beleuchtung und Verschattung nochmals in anderer Perspektive berechnet wird. Dies würde in den allermeisten Titeln viel zu viel Rechenleistung benötigen. Also greifen die Entwickler zu Tricks: Cube-Maps und Screen-Space-Reflections. Erstere können Sie sich in etwa wie eine Skybox vorstellen: Aus mehreren (vor-)berechneten Bildern wird ein Würfel kreiert. Der Inhalt dieses Quaders wird auf die spiegelnde Oberfläche gelegt und perspektivisch mit der Blickrichtung des Spielers synchronisiert.

Ändern Sie also den Blickwinkel, dreht sich der Würfel entsprechend, was eine ansatzweise glaubhafte Spiegelung ermöglicht – wenn man nicht genau hinsieht. Cube-Maps kommen schon lange und häufig in Spielen zum Einsatz. Prominentes Beispiel sind Rennspiele und der spiegelnde Lack der Fahrzeuge. Nachteile von Cube-Maps sind unter anderem die stark eingeschränkte Perspektiventreue, die geringe Auflösung und das Problem, dass eine Echtzeitberechnung teuer ausfallen würde.




Links: Der obere Teil ist eine Cube-Map-Reflexion
Rechts: Der Hinterkopf kann nicht gespiegelt werden

## Screen-Space-Reflections

Hier kommen Screen-Space-Reflections ins Spiel, die hochauflösende und perspektivisch glaubhaftere Reflexionen ermöglichen. Allerdings mit Einschränkungen, denn wie das Präfix „Screen-Space“ schon suggeriert, kann mit dieser Art Spiegelung nur der Screen-Space, also der Bildschirminhalt, berücksichtigt werden.

Prinzipiell ist diese Art Reflexion schon recht nahe dem Raytracing, doch das Verfahren ist stark vereinfacht. So suggeriert der Präfix „Screen-Space“, dass einzig der Bildschirminhalt berücksichtigt wird: Alles außerhalb des für den Spieler sichtbaren Bildes kann nicht gespiegelt werden. Das gilt beispielsweise auch für die eigene Spielfigur, weshalb man in First-Person-Shootern in der Regel nicht das alter Ego zu Gesicht bekommt, wenn man in einen Spiegel blickt. Es gibt seltene Ausnahmen, dann kommt jedoch auch eine andere Technik zum Einsatz. Oder das Charaktermodell wird für spezielle Spiegelungen gesondert berechnet – bei Mirror's Edge Catalyst und der Heldin Faith zum Beispiel. Bei ScreenSpace-Reflections werden zudem häufig temporale Verrechnungen genutzt, um Leistung zu sparen, Artefaktbildung zu kaschierern und Flimmern zu unterdrücken.

## Schwachstellen der Technik

In einer Bildbereichsreflexion ist also im Grunde nur das zu sehen, was gerade auf dem Bildschirm ist. Alles andere wird vom Shader ausgeblendet, was zu einigen sehr auffälligen und befremdlich wirkenden Nebeneffekten führt.

Zu weiteren Problemen kommt es, wenn sich vor der Spiegelung ein Objekt befindet, beispielsweise das Waffenmodell in einem Ego-Shooter. Denn zum einen befindet sich das Objekt im Weg: Bei der Berechnung der Reflexion wird ein unsichtbarer Strahl von dem virtuellen Auge des Spielers auf die spiegelnde Oberfläche geschickt. Das Waffenmodell würde diesen Strahl aufhalten. Es muss außerdem in fast allen Fällen ein großzügiger Bereich um das betreffende Objekt herum ausgeschnitten werden, damit dieses nicht optisch inkorrekt selbst in die Spiegelung gelangt.

## Hilfe dank Cube-Maps

Ein ziemlich abenteuerliches Beispiel liefert Battlefield 1. Der Stacheldrahtzaun vor der mit Screen-Space-Reflections verzierten Wasseroberfläche torpediert deren Darstellung stark. Es handelt sich jedoch um keinen Grafikfehler, an dieser Stelle limitiert die Technik. Um solcherlei Auffälligkeiten zu kaschieren, werden häufig Cube-Maps an den Stellen eingeblendet, an denen die Screen-Space-Reflections aussetzen.

In The Division funktioniert dies beispielsweise sehr überzeugend: Der Open-World-Shooter aktualisiert alle paar Frames die Cube-Maps mit neuen, aus der Umgebung und der vorherrschenden Lichtsituation gewonnenen Daten. Außerdem filtert der Ubisoft-Titel die Reflexionen sehr geschickt und vermischt sie temporal mit den Cube-Maps. Probleme mit Spiegelungen der Charaktere sind außerdem kaschiert und eher unauffällig.

## Idealfälle

Geschickt eingesetzt können auch getrickste Spiegelungen überzeugen. Es gibt einige Beispiele, in denen Screen-Space-Reflections wirklich klasse aussehen. Dazu zählt Doom, das sehr schicke Bildbereichs-Reflexionen bietet, diese sauber mittels einer temporalen Komponente inklusive Verwischschutz (Weighting) glättet und sie außerdem geschickt mit Environment-Cube-Maps vermischt. Auf diese Weise werden viele der Nebeneffekte der Screen-Space-Reflections kaschiert.

## Deus Ex: Mankind Divided

Die in Mankind Divided erstmals genutzte Dawn-Engine zeigt prinzipiell sehr schicke Screen-Space-Reflections, doch es gibt auch hier einige Auffälligkeiten.

Technisch fährt der auf der Glacier-2-Engine (Hitman) basierende Grafikmotor von Deus Ex einiges auf. Die schicken Spiegelungen werden in Mankind Divided freizügig eingesetzt und verstärken in dem Titel - ähnlich wie die Reflexionen in Mirror's Edge - den „cleanen" Sci-Fi-Look. Die qualitativ hochwertigen Screen-Space-Reflections werden mit Cube-Maps ergänzt, letztere sind allerdings recht niedrig aufgelöst. Zudem sind sie statisch und bilden nur unbewegte Objekte ab (Bild links unten, Punkt 1).

Die Spiegelungen zieren zudem auch vertikale Flächen, wir können an solch einer Wand einmal gut die typischen Probleme mit der Charakter-Darstellung demonstrieren. In Ego-Perspektive würde Jensen gar keine Spiegelung zeigen, weil die Figur nicht im Bildbereich zu sehen ist. In der Third-Person-Ansicht kann ebenfalls nur der Teil gespiegelt werden, der sich gerade sichtbar im Bild befindet (Bild links unten, Punkt 2).

## Fazit

Screen-Space-Reflections sind schick, aber zeigen praktisch immer irgendwelche unschönen Nebeneffekte und Artefakte. Sie kosten außerdem häufig viel Leistung, obwohl nur der Screen-Space in die Berechnung einbezogen wird. Dennoch sollte man nach Möglichkeit nicht auf sie verzichten, denn sie können durchaus den gesamten Look einer Szene beeinflussen.

# Screen-Space-Reflexionen

Screen-Space-Reflections zeigen häufig eine Reihe Artefakte, darunter Grieseln, welches auf die recht ungenaue Abtastung des Shaders und die Auflösung zurückfällt. Häufig wird dies durch eine temporale Filterung bekämpft. Die auffälligsten Nebeneffekte stammen jedoch von dem Umstand, dass nur der Bildschirminhalt (Screen Space) gespiegelt werden kann.

Stellen Sie sich das anhand der drei unteren Bilder in etwa so vor: Der Shader zeichnet eine virtuelle Linie (Tracer) von Ihrem Blickfeld auf die Pfütze. Von deren Oberfläche wird dieser Strahl im korrekten Winkel reflektiert, dann trifft er auf die Türe. Der Pixel in der Pfütze (der etwas gröber ausfällt, um Rechenzeit zu sparen), übernimmt nun einen Teil des Farbwerts des Punktes, auf den er traf. Das Gleiche gilt für die umliegenden Bildpunkte; zusammen entsteht so eine perspektivisch relativ korrekte und hochauflösende Echtzeit-Spiegelung. Allerdings funktioniert dies nur, wenn der Strahl nicht einen Punkt außerhalb des Bildes trifft, denn diese werden für die Szene nicht berechnet – es käme also eine leere Information zurück, die Reflexion bleibt aus. Ändern wir also den Blickwinkel, können wir die Reflexion „aus dem Bild wischen“. Und daher bleibt der Spiegel im ersten Bild leer. Der Bereich hinter uns existiert nicht im Screen-Space, ebenso wenig unsere Spielfigur.

Dieser Spiegel zeigt keine Spiegelung – weder der Bereich hinter uns noch unsere Spielfigur befinden sich im Screen-Space, sie können also auch nicht gespiegelt werden.

Noch wirkt die Spiegelung überzeugend.

Doch verschieben wir den Blickwinkel...,

...enthält der Screen-Space nicht ausreichend Daten.

Ähnlich gute und nochmals (im positiven Sinne) auffälligere Spiegelungen bietet The Division. Auch hier werden mehrere Ansätze genutzt. Die Screen-Space-Reflections fallen hochauflösend und präzise aus, werden sauber temporal gefiltert und zeigen kaum jene Artefakte, die in vielen Spielen auftauchen, wenn sich ein Objekt vor einer Spiegelung befindet: In diesem Fall muss das betreffende Objekt vom Shader ausgeschnitten werden, um nicht selbst erfasst und in der Spiegelung sichtbar zu werden. Sehr auffällig tritt dies in Ego-Shootern um das Waffenmodell oder in Third-Person-Perspektive um die Spielfigur auf. Im Normalfall ist ein deutlicher und hässlicher „Rand“ sichtbar, in dem die Reflexion aussetzt. Dies ist in The Division nur sehr eingeschränkt zu beobachten. Außerdem nutzt der Optikkracher sich alle paar Frames neu berechnende Cube-Maps, welche die Screen-Space-Reflections dort ergänzen, wo sie technisch aussetzen.

Eine fast perfekte Illusion: Die Kombination aus hochauflösenden und sauber gefilterten Screen-Space-Reflections und Environment-Cube-Maps in The Division.

Seltener Anblick: Eine Third-Person-Spiegelung – der Frontbereich bleibt allerdings leer.

Eine noch nicht vollständig aktualisierte Cube-Map-Reflection

Einige Frames später ist die Spiegelung neu berechnet und ausgetauscht.

# Moderne Grafikeffekte: Umgebungsverdeckung

Die realistische Umsetzung von Licht und Schatten gehört zu den anspruchsvollsten Rendertechniken in Computerspielen beziehungsweise Computer Generated Imagery (CGI).

## Echtzeit-Berechnung

Während bei gerenderten Filmen oder Bildern auf sehr komplexe Techniken zurückgegriffen werden kann und via Raytracing eine wirklichkeitsnahe Simulation von Licht und Schatten möglich ist, wäre dies in Spielen viel zu aufwendig für aktuelle Hardware. Um dennoch ein stimmiges Gesamtbild zu erhalten, nutzen Entwickler vereinfachte Raytracing-Annäherungen.

Ein solcher Rendering-Trick ist die Umgebungsverdeckung oder Ambient Occlusion. Dieser Post-Processing-Effekt simuliert den Umstand, dass in der Realität weniger Licht in Ecken und enge Spalten gelangt, da ein Teil der eintreffenden Strahlen zuvor von umliegenden Oberflächen reflektiert werden und daher nicht bis an diese Stellen vordringen. Ecken und Kanten wirken daher dunkler. Diese Lichtstrahlen (Rays), deren Wege (Traces) und Abpraller (Bounces) tatsächlich in Echtzeit zu berechnen, würde aktuelle GPUs komplett überfordern.

## Entwickler-Tricks

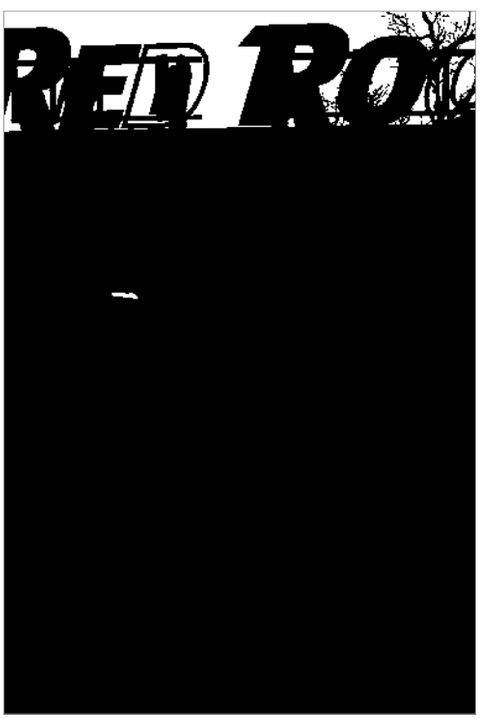
Seit einigen Jahren kommt daher in Spielen die Umgebungsverdeckung zum Einsatz. Dabei handelt es sich prinzipiell um ein Graustufen-Bild, das aus dem Tiefen-Buffer (auch Depth- oder Z-Buffer) generiert und im Anschluss über die Spielszene gelegt wird (Post-Processing).

In den Bildern unten auf dieser Seite sehen Sie die Spielszene ohne Umgebungsverdeckung dann den Depthbuffer, welcher die Tiefeninformationen für den zu berechnenden Effekt enthält und schließlich die Textur für die Umgebungsverdeckung. Während der Post-Processing-Phase wird diese mit dem ursprünglichen Bild vermischt (Weiß = transparent).

In den meisten Fällen handelt es sich bei der Ambient Occlusion zudem um einen Screen-Space-Effekt, es können und werden also nur Daten genutzt, die sich sichtbar im Bildbereich befinden, was zu einigen Nebeneffekten führen kann. Neben dem klassischen SSAO (Screen-Space-Ambient-Occlusion) ist auch Nvidias HBAO+ ein solcher Bildbereichseffekt. Außerdem sorgt der Umstand, dass es sich bei dem Post-Processing-Effekt um keine „echte“ dreidimensionale Berechnung handelt, sondern um eine aus 2D-Daten kreierte Textur, für einige Unstimmigkeiten.

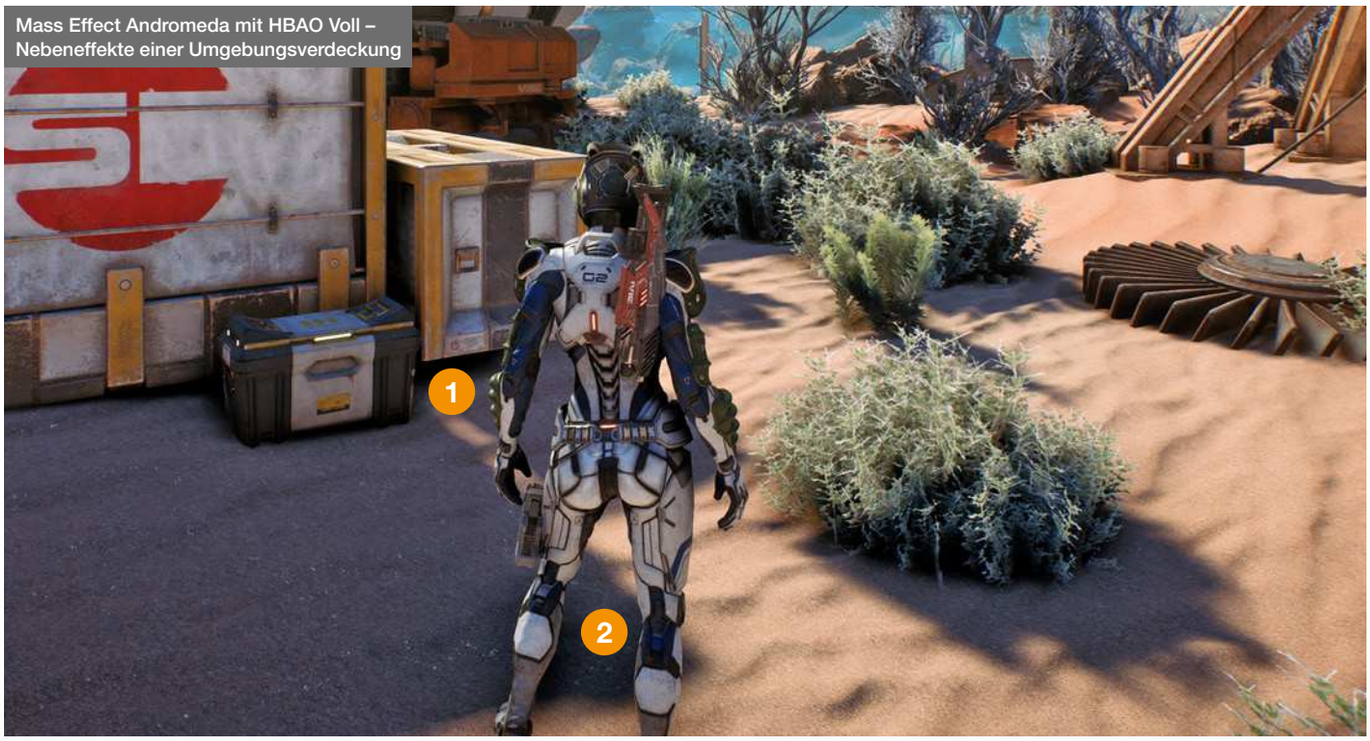
Auf den folgenden Seiten wollen wir einige Auffälligkeiten dokumentieren. Wir nutzen dafür mehrere Spiele, die auf eine Vielzahl verschiedener Techniken setzen.

Darunter das klassische SSAO, welches in The Witcher 3 zudem in einer niedrigeren Auflösung berechnet wird, die „freie“ HBAO-Variante von Mass Effect: Andromeda, sowie verschiedene Implementierungen von Nvidias HBAO+ und schlussendlich das auf 3D-Voxeln basierende VXAO, welches die nächste Evolutionsstufe des Effekts darstellt.

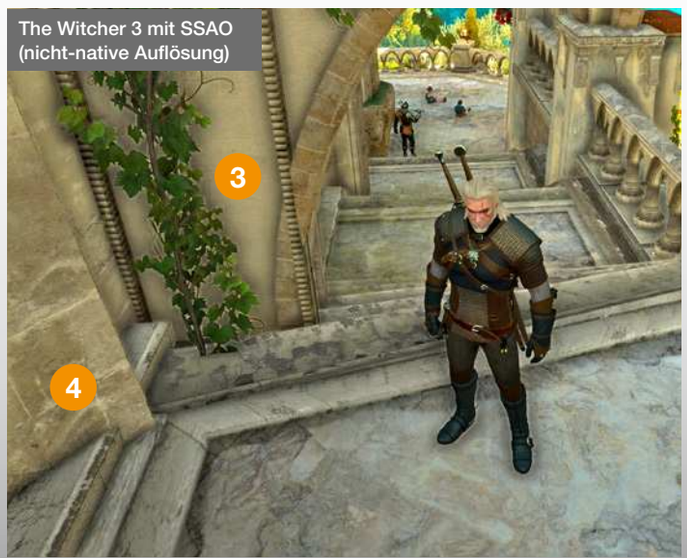


# Umgebungsverdeckung

Nicht selten fällt direkt ins Auge, dass die Umgebungsverdeckung sehr stark akzentuiert wird. Das wirkt zwar dramatisch und „hochqualitativ“, ist aber unrealistisch. (1) Da es sich bei der Umgebungsvereckung in Mass Effect: Andromeda um einen Screenspace-Effekt handelt, kann der Shader den Schatten unter der Kiste nicht vollständig berechnen; nötige Informationen werden von unserer Spielfigur verdeckt, der Schatten verschwindet. Aus diesem Grund ist um Objekte und Figuren eine Art „Halo“ oder „Schein“ zu erkennen, in dem die Umgebungsverdeckung aussetzt. (2) Achten Sie auf die regulären Schatten: Zum einen ist die Umgebungsverdeckung viel zu dunkel, außerdem kommt das Licht in dieser Szene von vorn – dieser Schatten wäre in der Realität unmöglich. (3) SSAO in The Witcher 3 wird nicht in voller Auflösung berechnet, daher deckt sich die gerenderte AO-Textur nicht 1:1 mit der Spielszene, es entstehen helle Ränder um Figuren und Objekte. HBAO+ wird in voller Auflösung berechnet und fällt deutlich weniger stark durch Artefakte auf. Außerdem tastet HBAO+ feiner ab und bezieht auch Normal-Maps mit in die Berechnung ein. (4) Diese Ecke wirft einen Schatten auf die Bordsteinkante und die Mauer darüber. Das wäre in der Realität nicht möglich.



Mass Effect Andromeda mit HBAO Voll – Nebeneffekte einer Umgebungsverdeckung



The Witcher 3 mit SSAO (nicht-native Auflösung)

The Witcher 3 mit HBAO+ (native Auflösung)

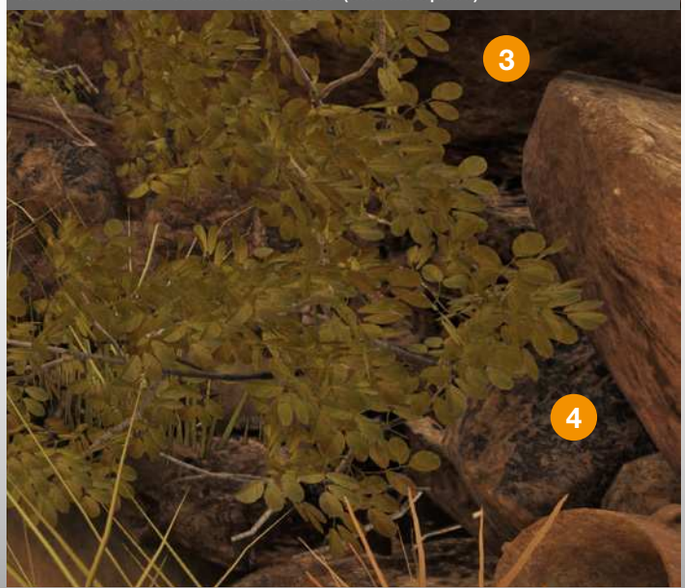
# Screen-Space vs. World-Space

Wie wir bereits im vorherigen Teil dieser Artikel-Reihe erörtert haben, kommen Screen-Space-Effekte mit einigen Nebeneffekten daher: Sie lassen sich beispielsweise mit dem Verändern des Blickwinkels „aus dem Bild wischen“. Dies gilt auch für Nvidias HBAO+, welches wir uns an dieser Stelle genauer ansehen wollen. Achten Sie einmal bei (1) auf den (relativ undeutlichen) Schatten des hölzernen Balkens rechts. Im ersten Bild ist wird die rote Mauer noch verdunkelt, doch bewegen wir unsere Spielfigur ein wenig nach links, verschwindet der Schatten plötzlich. Ein anderer häufig zu beobachtender Nebeneffekt ist der Schattenkranz um unser Waffenmodell bei (2). In Rise of the Tomb Raider können wir dank VXAO-Support vergleichen, wie sich eine Screen-Space-Berechnung von einer World-Space-Berechnung unterscheidet. Die sich bei (3) befindlichen Felsen werden nicht vollständig verschattet, da sich die Schatten werfenden Objekte nicht im Screen-Space befinden. Auch der Krug bei (4) befindet sich zu nah am Bildschirmrand. Bei VXAO handelt es sich zudem um einen voxelbasierten Effekt, daher wird auch das Geröll hinter dem Busch vollständig verschattet, während HBAO+ an diesen Stellen teilweise ausblendet. VXAO erfasst außerdem die Normal-Maps genauer. Allerdings ist der prinzipiell sehr schicke Voxeleffekt nochmals dunkler als das schon unrealistisch dunkle HBAO+; viel Leistung benötigt diese fortschrittliche Art der Umgebungsverdeckung außerdem.

HBAO+ in Fallout 4 – Unmögliche Verschattung um das Waffenmodell ...

... und verschwindende Schatten am Bildschirmrand

Rise of the Tomb Raider mit HBAO+ (Screen-Space)




Rise of the Tomb Raider mit VXAO (World-Space)

# Moderne Grafikeffekte: Texturdarstellung

Der Begriff „Textur“ dürfte den meisten Lesern bekannt sein. Die zweidimensionalen Tapeten begleiten die meisten Spieler seit Beginn ihrer Laufbahn. Doch die Einsatzgebiete sowie die Art und Weise, wie das Texturmodell mit der Beleuchtung in Spielen interagiert, haben sich stark gewandelt – Stichwort Physically Based Rendering.

## Mehr Realismus durch Wirklichkeitstreue

Kling logisch, ist aber so: Je näher ein Renderverfahren an die Realität angenähert wird, desto überzeugender wirkt die Spielegrafik auf das menschliche Auge. Um den Launch der aktuellen Konsolengeneration begannen die Entwickler, ihre Engines auf physikbasierte Rendertechnologien umzustellen.

Dabei hat Physically Based Rendering (PBR) keine fest vorgeschriebenen Regeln, sondern ist eher ein loses Konzept. Jeder Entwickler und jede Engine kann ein solches Konzept anders umsetzen. Schauen wir uns zwei fiktive Spiele mit PBR an, unterscheiden sich diese mit einiger Wahrscheinlichkeit optisch trotz des Wirklichkeitsbezugs der Rendertechnologien.

Prinzipiell funktioniert das Shadermodell der Texturen (Physically Based Shading, PBS) aber ähnlich: Gegenüber konventionellen Texturmodellen, bei denen ein vom Texture-Artist erstelltes Bild oder Foto einer Oberfläche samt einem Großteil statischer Beleuchtungs- und Verschattungsinformationen über ein Polygonnetz gelegt wird, bekommen PBS-Texturen physikalische Attribute zugewiesen.

Die drei Haupteigenschaften sind Albedo (Grundfärbung), Microsurface (Oberflächenbeschaffenheit) und Reflectivity (Reflexionsverhalten). Diese Eigenschaften formen ein PBR-Material, das sich je nach virtuellem Lichteinfall anders verhält und diesem anpasst. Dabei gilt teilweise der physikalische Energieerhaltungssatz. Eine spiegelnde Oberfläche etwa kann nicht heller strahlen als die eigentliche Lichtquelle – wobei dies nur für Materialien gilt, Post-Processing-Effekte wie Bloom oder Screen-Space-Reflections können dies torpedieren.

## Winzig und doch so wichtig

Die Rolle der Microsurfaces ist dabei eine sehr wichtige. So bestimmen sie, wie stark eintreffendes Licht gestreut wird. Eine raue Oberfläche streut das Licht stärker, das Material wirkt stumpfer. Diesen Effekt können Sie auf der nächsten Seite gut anhand des Matsches von Battlefield 1 erkennen.

Dieser besteht aus eine Vielzahl dieser Microsurfaces, welche dem Schlamm unterschiedliche Reflexionseigenschaften verleihen, obwohl prinzipiell sehr ähnliche Materialien zum Einsatz kommen. Dies steigert die Dynamik der Oberfläche, sie ist kaum noch als eine Textur identifizierbar.

Im weiteren Sinne der Microsurfaces nutzt Battlefield 1 zudem Displacement-Mapping und Tessellation, um den Oberflächen zusätzliche Tiefe zu verleihen; aus Textur wird Geometrie.

## Parallax- vs. Displacement-Mapping

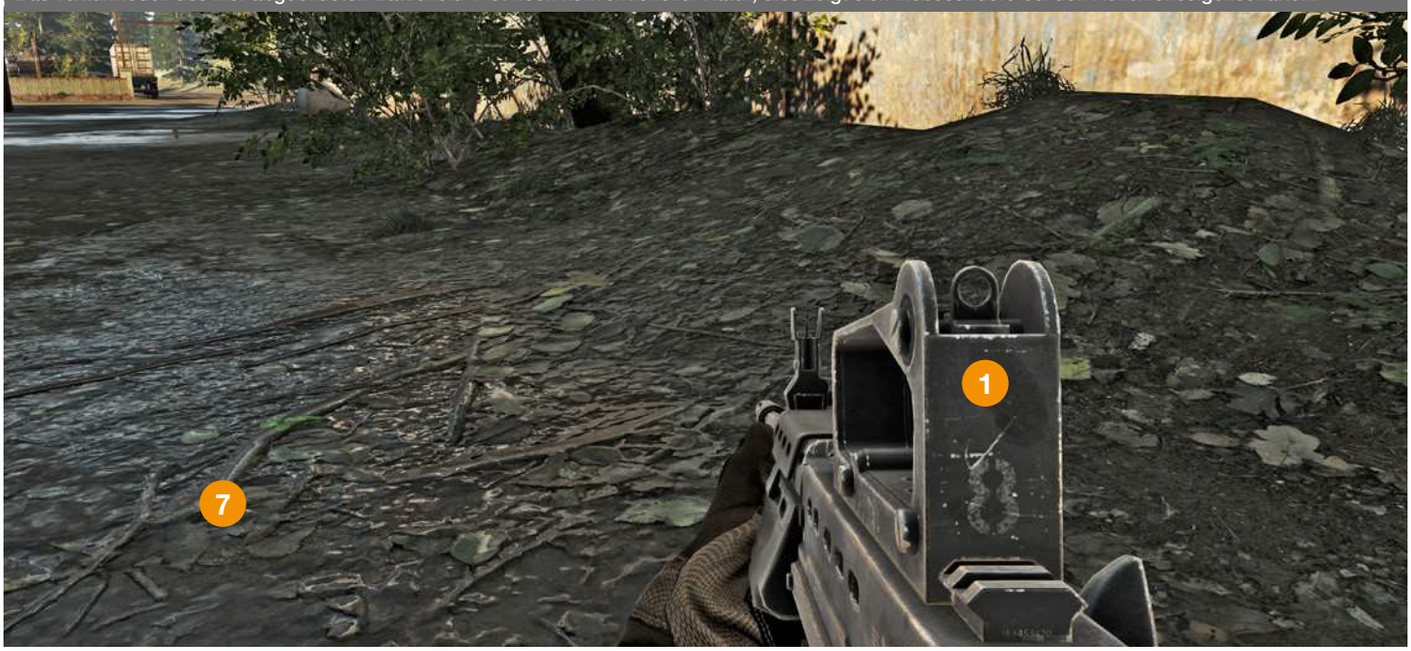
**Die beiden Technologien sind nicht immer einfach zu unterscheiden. Wir zeigen, wo die größten optischen Unterschiede liegen.**

**Parallax-Occlusion-Mapping ist eine interessante Technik, denn anders als beim Displacement-Mapping, welches zusätzliche Geometrie erzeugt, bleibt die Fläche zweidimensional. Dieser Umstand macht es auch möglich, die beiden Techniken zu unterscheiden. Blickt man in flachem Winkel auf eine betreffende Oberfläche oder nutzt eine Kante, um beispielsweise eine Mauer zu untersuchen, zeigt sich die Zweidimensionalität einer Parallax-Map. Kommt stattdessen eine Displacement-Map zum Einsatz, wie bei der Bodentextur im ganz rechten Bild, lässt sich im flachen Winkel zweifelsfrei die zusätzliche Geometrie erkennen. Diese Technik kann allerdings auch reichlich Leistung kosten, da die Grafikkarte viele zusätzliche Polygone berechnen muss, insbesondere wenn der Detail- beziehungsweise Tessellationsgrad hoch ausfällt. Die Parallax-Map kostet nur wenig Leistung und etwas Speicher.**

# Texturen und Oberflächenveredelung

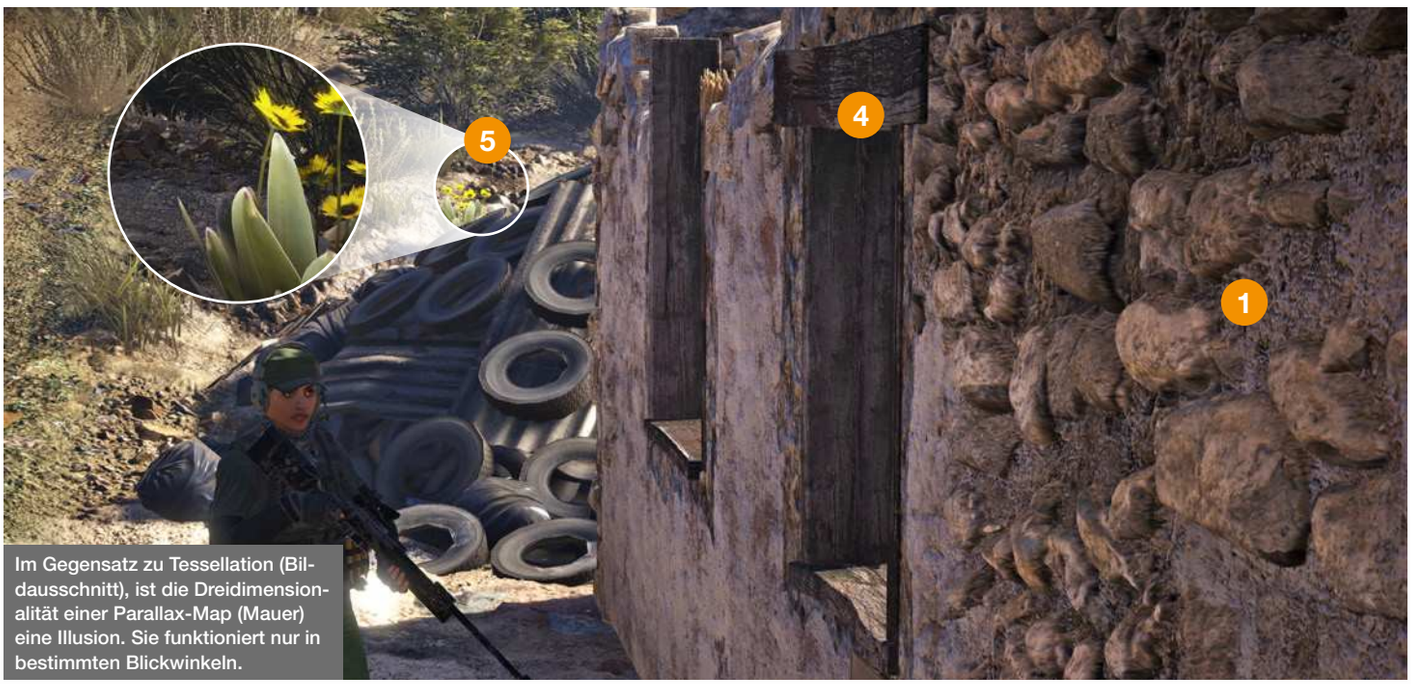
Konventionelles Texturmodell versus physikbasiertes Modell (PBR): (1) Das konventionell texturierte Waffenmodell in Battlefield 4 überzeugt in dieser Beleuchtungssituation nicht wirklich, sondern wirkt in dieser Szene wie ein Fremdkörper. (2) Die Materialien des Battlefield-1-MGs hingegen greifen das eintreffende virtuelle Licht auf und streuen sowie reflektieren die Strahlen je nach Oberflächenbeschaffenheit: Das Metall des Zielfernrohrs (3) ist augenkundig ein anderes als jenes, aus dem das Magazin gefertigt wurde. Im direkten Vergleich sind auch die Reflexionseigenschaften der Bodentexturen sehr auffällig. Mit PBR wird verstärkt auf Microsurfaces gesetzt, eine große Fläche wird also aus vielen kleinen zusammengesetzt, deren Materialeigenschaften sehr unterschiedlich ausfallen können. Dies lässt sich gut anhand des Battlefield-1-Matsches erkennen: Achten Sie einmal auf den eher stumpf glänzenden Schlamm in den Abdrücken der Panzerketten (4) und vergleichen Sie ihn mit den hell glänzenden Fußabdrücken (5). Darüber hinaus ist der Fresnel-Effekt eines physikbasierten Renderers zu erkennen: Je flacher der Blickwinkel auf eine Oberfläche fällt, desto stärker reflektiert sie. Battlefield 1 erzeugt zudem zusätzliche Details via Displacement-Maps und Tessellation – tatsächliche Geometrie also (6). Battlefield 4 nutzt hingegen nur Normal-Mapping, um bei Texturen eine Tiefe zu suggerieren (7).

Das Texturmodell des hier abgebildeten Battlefield 4 ist noch konventioneller Natur; dies zeigt sich insbesondere bei den Reflexionseigenschaften.

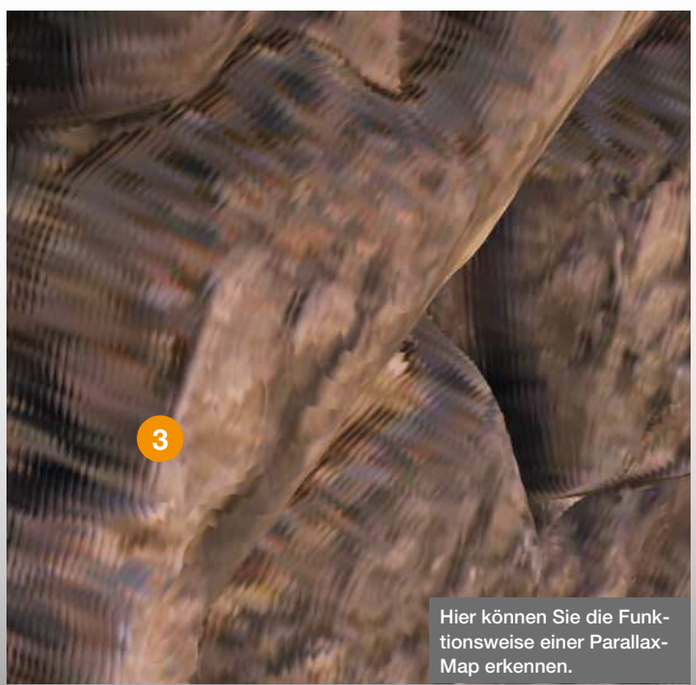


Neben einem physikbasierten Renderer, der die Reflexionseigenschaften realitätsnah simuliert, nutzt Battlefield 1 zudem Tessellation, um Oberflächen zu veredeln.

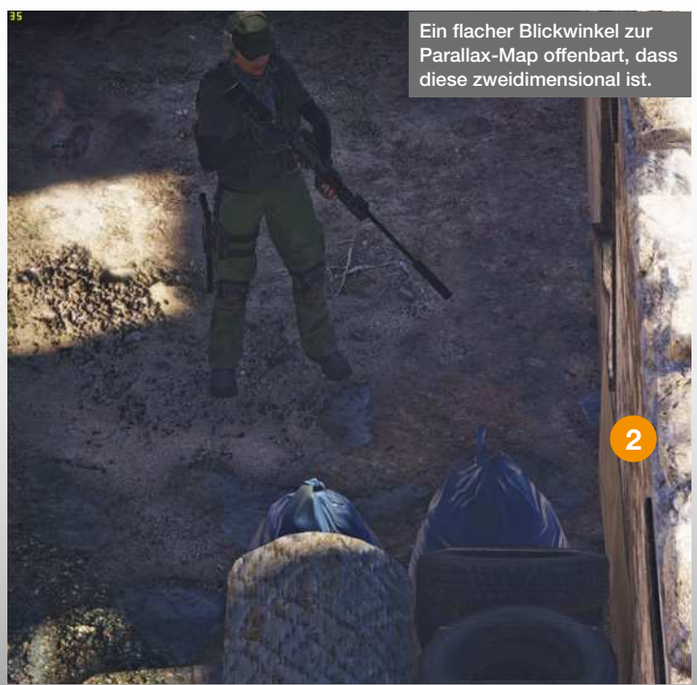
Oberflächenveredelung via Tessellation und Parallax-Occlusion-Maps: Der steigende Detailgrad moderner Spiele erfordert schon seit geraumer Zeit mehr als bloß flache Texturen. Seit Jahren ist daher zuerst Bump-, dann Normal-Mapping Bestandteil vieler Pixeltapeten. Diese Maps enthalten Höheninformationen in Form einer zusätzlichen Schwarzweiß- (Bump-Map) oder farbigen Dreikanal-RGB-Textur (Normal-Map). Letztere wird zumeist aus einem zugrundeliegenden, sehr hochauflösenden Polygonmodell generiert und auch dazu genutzt, bei 3D-Modellen einen höheren Polygoncount zu suggerieren. Das nochmals aufwendigere Parallax-Mapping wirkt sehr plastisch (1), ist jedoch noch immer eine 2D-Textur (2). Bei dieser Technik wird die Textur mehrfach übereinander gelegt, dann werden die Texturkoordinaten der einzelnen Pixeltapeten in Abhängigkeit des Blickwinkels verschoben, was zu einem dreidimensionalen Effekt führt. Dabei entsteht jedoch auch eine Art „Treppchenbildung“ (3). Zudem wird beim Bewegen des Blickwinkels ein „Wabern“ sichtbar, wenn die Texturen verschoben werden. Obendrein gibt es Unstimmigkeiten bei der Perspektive: Achten Sie einmal auf den Balken am Fenster, es wirkt als schwebe dieser (4). Die „Königsklasse“ dieser Techniken ist das Displacement-Mapping, welches eine Height-Map nutzt und mittels dieser (häufig via Tessellation) echte Geometrie erzeugt (5).




Im Gegensatz zu Tessellation (Bildausschnitt), ist die Dreidimensionalität einer Parallax-Map (Mauer) eine Illusion. Sie funktioniert nur in bestimmten Blickwinkeln.




Hier können Sie die Funktionsweise einer Parallax-Map erkennen.




Ein flacher Blickwinkel zur Parallax-Map offenbart, dass diese zweidimensional ist.

# Moderne Grafikeffekte: Kantenglättung

Mit allem, was die moderne Technik hergibt (hier: Ultra-HD-Auflösung mit SMAA sowie SSAA), verringern sich selbst kleinere Kanten in den kaum noch wahrnehmbaren Bereich – mit ihnen aber leider auch die Framerate …

Selbst 4K ist nicht genug: In diesem Shot nutzen wir neben SMAA das sehr ineffiziente Supersampling

Eines der ältesten Probleme der Spielegrafik sind flimmernde Kanten – bis heute. Trotz immer höherer Auflösungen bleiben die Nebeneffekte der Rastergrafik bestehen, es kommt zu Pixelflimmern oder fachlich korrekter: Aliasing.

## Wie Aliasing entsteht

Aliasing tritt auf, da Inhalte, die mit einem Rasterizer gerendert werden, in ein Pixelraster passen müssen. Dieses Raster beschreibt im Normalfall die gewählte Auflösung, mit der die Grafik berechnet wird. Einfach formuliert: Je gröber die Auflösung und damit das Raster und je feiner der darin dargestellte Inhalt, desto flimmeranfälliger wird das Gesamtbild.

Noch deutlicher wird Aliasing, wenn das Bild nicht statisch ist, sondern dem Wandel unterworfen ist – ein bei Spielen sehr häufiger Umstand. Bewegt sich der Bildinhalt, beispielsweise durch eine Eingabe des Spielers, so verschiebt sich auch der Inhalt des Rasters. Je nach Blickwinkel, Entfernung und Bewegung kann dies wiederum dazu führen, dass Elemente, die zuvor in das Raster passten, nun zu fein ausfallen und verschwinden, während andere Feinheiten, die zuvor durch das Raster fielen, dank der Repositionierung nun angezeigt werden können. Dieses An- und Ausflackern der Pixel bei Bewegung ist besonders auffällig und störend.

Besonders betroffen sind neben Polygonkanten Texturen und Shader, die deutlich komplexer ausfallen, als noch vor einigen Jahren. Die Pixeltapeten bestehen heute nicht mehr aus einer einfachen Bitmap, sondern werden aus vielen einzelnen Maps zusammengesetzt. Darunter fallen Normal-Maps für Tiefeninformationen, Specular-Maps für Glanz und Highlights und Alpha-Test-Maps für Transparenzen. All diese Elemente neigen in der Praxis zu starkem Flimmern – und aufgrund der Feinheit der in Bewegung hochfrequenten Inhalte stört dies selbst bei Auflösungen wie Ultra HD und gar darüber noch. Besonders Alpha-Test-Texturen, die häufig für filigrane Inhalte wie Vegetation, Maschendrahtzäune oder Haare zum Einsatz kommen, sind problematisch.

## Ein Blick zurück

Schon vor gut 20 Jahren wurden Kantenglättungsverfahren eingesetzt. Lange Zeit sehr populär war dabei die Multisampling-Kantenglättung. Diese wurde entwickelt, um das wesentlich teurere Supersampling abzulösen. Das lateinische „Super“ bedeutet dabei „von oben herab, über“, „Sampling“ steht für „Abtastung“. Das Bild wird also von einem höheren Informationsgehalt auf einen niedrigeren heruntergerechnet, jeder Bildpunkt enthält nach dem Komprimieren des Inhalts Informationen von mehreren benachbarten Pixeln. Beim Herunterrechnen gehen zwar einige Feinheiten verloren, trotzdem enthält das Bild mehr Informationen als ein in nativer Auflösung berechnetes.

## Vereinfachte Variante

Beim Multisampling werden indes nicht alle Pixel mehrfach abgetastet, sondern nur jene, die an Polygonkanten liegen. Alle anderen Pixel werden nur einmal abgetastet. Damit bleiben aber auch die restlichen Inhalte, darunter Texturen, Shader und die häufig besonders flimmeranfälligen Alpha-Tests vom regulären Multisampling aus-

geschlossen und müssen aufwendig zusätzlich inkludiert werden (zum Beispiel per Alpha-to-Coverage-Filterung, Transparenz-Supersampling).

## Höhere Polygondichte

Modernere Titel haben nicht nur eine erhöhte Polygondichte, sondern zudem feine Inhalte abseits der Polygonkanten, welche Multisampling nicht erfasst. Zwar gibt es die Möglichkeit, MSAA zu erweitern, etwa durch Transparenz-Multisampling, doch ein solches erhöht auch die Leistungskosten und den Speicherbedarf.

Trotz dieser Erweiterungen lieferte das Multisampling ab etwa dem Jahre 2007 in vielen Fällen nicht jene befriedigende Glättung, die Spieler von früheren Titeln gewohnt waren. Zudem stieg der Performance-Hunger der ehemals günstigen Kantenglättung steil an – was den Untergang von MSAA einleitete, der bis heute anhält. Daran tragen vor allem aktuelle Rendertechniken Schuld.

## Deferred Rendering und Multisampling

Verstärkter Einsatz von Post-Processing-Effekten, insbesondere aber Deferred Rendering torpedieren den Einsatz von MSAA in aktuellen Spielen. Beim Deferred Rendering werden die Berechnungen unter anderem von Geometrie und Beleuchtung voneinander getrennt und in eigene Puffer gespeichert.

Einzelne Komponenten des Bildes können simultan berechnet werden, was tendenziell die Auslastung der Hardware und damit deren Effizienz erhöht. Erst im Anschluss wird aus mehreren, von der GPU berechneten Einzelbildern, darunter Tiefe (Depth), Ausrichtung (Normals) und Farbe (Albedo), dazu Licht- und Schattenberechnungen und eventuell zusätzlichen Komponenten, beispielsweise für das Post-Processing, das fertige Bild erstellt.

Dieses Renderverfahren ist inkompatibel zu MSAA, welches im Grunde jeden dieser Bildpuffer einzeln abtasten müsste, um die korrekten Farbwerte für die zu glättenden Pixel zu ermitteln. Dieser Umstand erhöht die Performance-Kosten sowie den Speicherbedarf von MSAA drastisch, häufig setzt das Multisampling in Kombination mit Deferred Rendering auch aus oder führt zu Artefakten.

Der Siegeszug des Deferred Rendering ist also neben der komplexer gewordenen Grafik und den feineren Inhalten außerhalb der Polygonkanten ein Grund für das langsame Aussterben des Multisampling-Antialiasings.

## Von Post-Process-AA zu TAA

Die Spieleschmieden setzten viele Jahre fast ausschließlich auf Post-Process-Antialiasing, um das Bild zu glätten, darunter neben MSAA etwa auch das wenig beliebte, weil die Bildschärfe stark beeinträchtigende FXAA. PC-Spieler mit hohem Qualitätsanspruch investierten indes überschüssige Leistung in optionales Super- oder Downsampling. Fortschrittliche Render-Techniken, darunter Physically Based Rendering und die technischen Fähigkeiten – aber auch deren Schwächen – der aktuelle Konsolengeneration, ließen aber bald darauf langsam eine neue Art Kantenglättungsverfahren entstehen.

## Temporale Kantenglättung

In den vergangenen Jahren gab es einen deutlichen Trend: Beinahe alle Entwickler bauen aktuell temporale Komponenten in ihre Engines ein. Diese nutzen mehrere hintereinander verrechnete Frames, um zusätzliche Bildinformationen zu sammeln; dies dient häufig mehreren Zwecken: So können beispielsweise grieselige Screen-Space-Reflections oder die Umgebungsverdeckung über mehrere Frames generiert werden. Durch Akkumulation – also eine Verdichtung des Effekts – wird ein noch besseres, höherauflösenderes und saubereres Ergebnis erzielt.

# Anti-Aliasing-Techniken (Auswahl)

| Kürzel | Bedeutung | Merkmale/Funktion |
|---|---|---|
| **FSAA** | Full-Screen/Scene-Anti-Aliasing | Allgemeiner Begriff für Vollbild-Anti-Aliasing |
| **SSAA** | Super-Sampling-Anti-Aliasing | Überbegriff für mehrfaches Sampling aller Pixel |
| **OGSSAA** | Ordered Grid Super-Sampling-AA | SSAA mit geordnetem Abtastmuster |
| **SGSSAA** | Sparse Grid Super-Sampling-AA | SSAA mit gestreutem Abtastmuster |
| **SMAA** | Enhanced Subpixel Morphological Anti Aliasing | Extrem teures, kaum nutzbares Supersampling |
| **SMAA 2Tx** | Enhanced Subpixel Morphological Anti Aliasing 2Tx | Erweiterte Version des morphologischen SMAA-Post-Processings |
| **MSAA** | Multi-Sampling-Anti-Aliasing | Überbegriff für mehrfaches Sampling von Polygonkantenpixeln |
| **SGMSAA** | Sparse Grid Multi-Sampling-AA | MSAA mit gestreutem Muster – ab etwa 2006 lange Zeit Quasi-Standard |
| **TMSAA** | Transparency Multi-Sampling-AA | Nvidias MSAA für transparente Texturen |
| **TSSAA** | Transparency Super-Sampling-AA | Nvidias SSAA für transparente Texturen |
| **AAA** | Adaptive Anti-Aliasing | AMDs adaptives AA für transparente Texturen |
| **CSAA** | Coverage Sample Anti-Aliasing | Nvidias Erweiterung von MSAA |
| **EQAA** | Enhanced-Quality Anti-Aliasing | AMDs Erweiterung von MSAA (erst seit der HD-6900-Reihe verfügbar) |
| **CFAA** | Custom Filter Anti-Aliasing | AMDs verbesserte Downfilter für glattere Bilder; fungiert als „Aufsatz“ für MSAA oder SSAA (verfügbar ab HD 2000) |
| **MLAA** | Morphological Anti-Aliasing | AMDs Post-Processing-AA in der Radeon-Software |
| **FXAA** | Fast Approximate Anti-Aliasing | Günstiges, kontrastbasiertes Post-Process-AA |
| **SRAA** | Subpixel Reconstruction AA | Fortschrittliches Post-AA, derzeit aber nirgendwo im Einsatz |
| **EER** | Edge-Equivalent Resolution | Qualitätsmerkmal von FSAA (mehr = besser) – aber kein eigenes AA! |
| **TAA** | Temporales Anti-Aliasing | Überbegriff für eine Glättung mit Hilfe mehrerer, aufeinanderfolgender Bilder |
| **TXAA** | Temporal Approximate AA | Shader-basiertes Verfahren von Nvidia |
| **DLSS** | Deep Learning Super Sampling | KI-gestützte Kantenglättung von Nvidia; wahlweise interne oder externe Hochrechnung („Upscaling“) von Auflösungen |

## Multi-Sample-AA und Forward-Plus-Rendering

**Noch gibt es Spiele, welche MSAA unterstützen. Insbesondere in einigen Rennspielen ist die altehrwürdige Hardware-Kantenglättung noch häufiger aufzutreffen.**

**Diese, darunter etwa Codemasters' Dirt-Reihe oder Forza Horizon 3 und 4 setzen in solchen Fällen jedoch häufig auf einen Forward-Plus-Renderer, eine Erweiterung des klassischen, für viele aktuelle Render-Techniken aber ungeeigneten Forward-Renderers.**

**Mit Forward-Plus-Rendering werden einige der größten Schwachstellen des Forward-Renderings umgangen, darunter die komplexe Beleuchtung mit mehreren dynamischen Lichtquellen. Doch für viele moderne Effekte, Render- und Beleuchtungstechniken ist auch Forward Plus nicht wirklich gut geeignet. Es ist also nur eine Frage der Zeit, bis auch die letzten Spiele auf das vielseitigere und bei komplexerer Beleuchtung potenziell auch effizientere und genauere Deferred-Rendering umsteigen.**

**Forza Horizon 4 nutzt dank Forward-Plus-Rendering noch eine gut greifende und relativ effiziente Multisampling-Kantenglättung samt Alpha-to-Coverage für Transparenzen**

RotTR nervte noch mit extrem flimmeranfälligem Content sowie unzureichender FXAA-, SMAA- und maßlos teurer SSAA-Kantenglättung, das jüngste Tomb Raider setzt auf TAA

Besonders interessant für Spieler ist die Temporale Kantenglättung (TAA). TAA existiert in vielfältiger Form und kann sich von Titel zu Titel unterscheiden. Das Prinzip ist aber immer ähnlich: Aus mehreren Frames werden Informationen gewonnen und so verrechnet, dass sie eine auffällige Pixelation verhindern. Dabei wirkt ein TAA auf das gesamte Bild, jeder Pixel wird durch die zeitliche Komponente mehrfach abgetastet („sampled"), weshalb einige temporale Kantenglättungen auch als Temporales Supersampling bekannt sind. Es handelt sich bei dieser Bezeichnung aber nicht um klassisches SSAA mit temporalem Ansatz, die tatsächliche Renderauflösung pro Frame ist nicht erhöht.

TAA hat allerdings auch einige handfeste Nachteile, so sorgt das Verrechnen über die Zeit für eine mehr oder minder starke Unschärfe und kann insbesondere bei niedrigen Bildraten mit Schlierenbildung beziehungsweise Ghosting deutlich nerven.

### TAA in aktuellen Spielen

Wie sehr ein TAA die grafischen Inhalte in Spielen beinträchtigen kann, lässt sich gut anhand des jüngsten Lara-Croft-Abenteuers Shadow of the Tomb Raider erahnen. Der Vorgänger setzte noch auf FXAA, SMAA (extrem teures, kaum nutzbares Supersampling) und war aufgrund feiner, häufig kontraststarker Inhalte extrem flimmeranfällig. Der Nachfolger ist ohne AA mindestens genauso flimmrig, ja neigt aufgrund häufig auftretender Dithering-Artefakte insbesondere bei Laras dynamischer Haarpracht vielleicht sogar zu noch stärkerem Flimmern und ist ohne Kantenglättung kaum genießbar.

Allerdings setzt Shadow of the Tomb Raider auf zwei unterschiedliche Varianten Temporaler Kantenglättungen, neben TAA auch auf eine mit zeitlicher Komponente erweiterte Version des morphologischen SMAA-Post-Processings, welches in Form von SMAA2Tx auch in anderen Spielen Verwendung findet. Interessanterweise gibt es neben dem regulären SMAA mit SMAA4x auch einen Ansatz, der die Post-Processing-Kantenglättung mit MSAA verbindet. Allerdings bleibt SMAA4x in Sachen Effizienz klar hinter dem angebotenen TAA und SMAA2Tx zurück. Shadow of the Tomb Raider setzt – wie übrigens viele aktuelle Spiele – offensichtlich einen derart starken Fokus auf die temporale Komponente und ihre Fähigkeiten, dass ein Betrieb ohne Temporales Antialiasing kaum mehr sinnvoll erscheint – oder gar von den Entwicklern gar nicht mehr gestattet wird.

### Temporal-AA als Standard?

Mit solch einem Fokus ist das jüngste Tomb Raider nicht allein. So lässt sich das TAA in Star Wars Battlefront 2 und Battlefield 5 nur noch in zwei Stufen wählen, nicht aber mehr deaktivieren. Und auch viele andere Titel setzen TAA für ein flimmerarmes Bild beinahe schon voraus – selbst in hohen Auflösungen.

Ein weiteres Beispiel ist Assassin's Creed Odyssey. Bereits seit geraumer Zeit verwendet Ubisoft ausgiebig Dithering für Überblendungen beim Level-of-Detail, aber auch für einige Transparenztexturen, darunter Haar oder halbtransparente Objekte wie Segel oder dekorative Stoffbahnen. Das TAA wirkt bei den pixeligen Transitionen bei Detailwechseln oder bei der Darstellung von Haaren – ähnlich wie bei Laras Haaren im Beispiel von Shadow of the Tomb Raider – ein wenig wie eine Rekonstruktionstechnik und sorgt dank der zeitlichen Komponente für ein deutlich saubereres,

ruhigeres und homogeneres Bild. Die ohne Kantenglättung extrem grieselig und unruhig wirkende Optik erhält erst durch das TAA den letzten Schliff.

## Quasi-Standard

Es ist allein wegen dieser Umstände nicht verwunderlich, dass beinahe jedes aktuelle Spiel auf eine temporale Kantenglättung setzt. Die feinen Inhalte, hohe Detailgrade und Sichtweiten sowie detaillierte Texturen und Modelle begünstigen allesamt Aliasing, welches selbst in hohen Auflösungen sehr störend ausfallen kann.

Des Weiteren kommt TAA auch zur Berechnung von einigen Post-Process-Effekten wie Screen-Space-Reflections und zur Kaschierung von Dithering-, aber auch Checkerboarding und anderen Upscaling-Artefakten zum Einsatz.

Letzterer Umstand ist ein weiterer Punkt, der TAA in vielen aktuellen Titeln im Grunde unverzichtbar macht: Die aktuelle Konsolengeneration hat feste Auflösungen weitgehend abgeschafft. Statt 1080p, 1440p oder 2160p nativ anzuzeigen, wird die Auflösung in modernen Konsolenspielen häufig dynamisch angepasst. Dieser Umstand hat sich nach dem Erscheinen der überarbeiteten Sony- und Microsoft-Geräte PS4 Pro und Xbox One X nochmals verstärkt. Werden diese Konsolen an ein 1080p-Display angeschlossen, bieten sie oft die Möglichkeit, das Full-HD-Bild per dynamischem Supersampling aufzuwerten.

## TAA als Bildverbesserung

Bei Konsolen kann ein entsprechend implementiertes TAA dynamisches Upscaling ermöglichen, die Technik funktioniert aber grundsätzlich auch andersherum. Anstatt also bei anspruchsvollen Szenen und zu wenig GPU-Leistung die Auflösung dynamisch zu reduzieren, ist es auch möglich, die Auflösung dynamisch zu erhöhen, falls die Bildrate hoch genug ausfällt und die Grafikkarte noch über Reserven verfügt.

Einige aktuelle Spiele unterstützen bereits eine optional dynamische Renderskalierung, die bei ausreichender hoher Performance über die gewählte Auflösung hinaus gehen kann, um so das Bild dynamisch zu verbessern. Titanfall 2 unterstützt beispielsweise ein TAA mit dynamischem Supersampling, welches – bei ausreichend GPU-Power – eine Auflösung bis 16K erreichen kann. Ubisoft bietet in AC Origins und Odyssey ebenfalls eine solche Option, versteckt sie allerdings ein wenig.

Allerdings sind Spiele mit dynamischem Supersampling auf dem PC noch relativ selten. Viel häufiger ist eine auf einen Wert festgelegte Ren-

# Kantenglättung bei Tomb Raider

**Ohne Kantenglättung ist Shadow of the Tomb Raider sehr flimmeranfällig. Dies geht nicht nur auf feine Inhalte zurück, sondern auch auf das genutzte Dithering, das bei LoD und einigen Transparenzen, darunter Laras Haaren eingesetzt wird. Eine temporale Kantenglättung wirkt bei diesen pixeligen Elementen ähnlich einer Rekonstruktionstechnik und sorgt für ein deutlich homogeneres Bild. SMAA mit MSAA wirkt etwas schärfer als TAA, glättet aber weniger effektiv und kostet Leistung sowie zusätzlichen Speicher.**

Full HD, keine Kantenglättung

Full HD, keine Kantenglättung

Full HD, SMAA

Full HD, TAA

Full HD, SMAA2Tx

Full HD, SMAA4x

Final Fantasy 15 ist der erste Titel mit Nvidas KI-befeuerten Deep Learning Supersampling (DLSS). Das Ergebnis überzeugt, wirkt optisch und artistisch aber leicht verändert

derskalierung vorgegeben, häufig geht diese auch nur von 100 Prozent ausgehend nach unten. Spiele auf dem PC unterstützen demnach häufiger Upscaling denn Downsampling, ein Umstand, der ob der hohen Leistung von PC-Komponenten etwas befremdlich wirkt. PS4 Pro und Xbox One X sind da (zumindest bei Full-HD-Darstellung) schon etwas weiter.

## TAA mit KI-Unterstützung

Am Horizont zeichnet sich bereits eine neue, sehr vielversprechende Technik ab. Mittels KI-Berechnungen können niedrig aufgelöste und pixelige Bilder auf eine überzeugende Art verbessert werden, selbst Hobby-Entwickler und Modder setzen bei der Aufbereitung von Grafiken bereits auf solch schlaue Bildverbesserer (siehe Kasten auf der rechten Seite).

## Deep Learning-Super-Sampling

Mit der Turing-Architektur und deren Tensor-Kernen stellte Nvidia indes Deep Learning Super Sampling vor. Dabei handelt es sich im Grunde um zwei verschiedene Ansätze: DLSS2x ist dazu gedacht, die GPU ein Spiel mit der gewählten Auflösungsstufe berechnen zu lassen, die Tensorkerne und KI-Algorithmen werden anschließend dazu genutzt, um das Bild intelligent zu verbessern. Diese Form von DLSS ist aktuell jedoch noch in keinem Spiel nicht verfügbar.

Die andere Form von DLSS ist ein KI-unterstütztes Upscaling. Mit diesem wird die Auflösung gegenüber der im Grafikmenü gewählten reduziert, das Bild dann hochgerechnet und von der KI verbessert. Diese Art von DLSS können wir uns bereits in einem Spiel zu Gemüte führen: Final Fantasy 15 unterstützt DLSS. Allerdings gibt es dabei eine starke Einschränkung, denn erstens funktioniert DLSS nur auf Turing-GPUs mit Tensor-Kernen, zweitens steht bislang nur eine Auflösung zur Verfügung. Nur wenn Sie mit einer passenden GPU im Optionsmenü 3.840 × 2.160 Pixel auswählen, ist DLSS verfügbar.

## Supercomputer im Hintergrund

Die eingeschränkte Wahl an Auflösungen ist dem Umstand geschuldet, dass Nvidia Supercomputer und Server mit entsprechenden Daten füttern muss, um die KI zu „lehren“. Für jedes Spiel und jede Auflösung müssen also separate Datensätze angelegt werden. Dies dürfte auch bedeuten, dass Nvidia den Support für ein Spiel oder eine spezifische Auflösung jederzeit einstellen und DLSS damit unbrauchbar werden könnte. Es bedeutet darüber hinaus, dass Nvidia für jedes Spiel und jede Auflösung eine Investition in Form von Rechen- und Serverzeit vorauszahlen muss – ein flächendeckender Support für zukünftige Spiele scheint allein deshalb wenig wahrscheinlich.

Neben Final Fantasy 15 ist jedoch bereits Unterstützung für mehrere Dutzend Titel angekündigt, darunter befindet sich auch Battlefield 5, welches mit seiner Raytracing-Implementation als besonders geeignet für das leistungssparende KI-Upscaling erscheint.

## Schick, aber noch nicht perfekt

Aktuell lassen sich die Auswirkungen von DLSS wie erwähnt nur in einem Spiel beobachten. Das mit sämtlichen (Nvidia-)Effekten extrem anspruchsvolle Final Fantasy 15 ist ein dankbares Demonstrationsobjekt, da die stark flimmeranfällige Grafik von einem eher unbefriedigenden TAA geglättet wird. Selbst in Ultra HD lässt dieses das Bild grob,

## TAA bei Odyssey aktivieren

**Assassin‘s Creed Odyssey setzt beinahe zwingend auf TAA. Feine Inhalte sowie das LoD und Transparenzen mit auffälligen Dithering-Artefakten machen dies fast zur Notwendigkeit. Ubisoft bietet glücklicherweise die Option eines temporalen Anti-Aliasings mit dynamischem Supersampling; diese ist allerdings ein wenig versteckt.**

**Zuerst müssen Sie Ihre gewünschte Auflösung wählen, dann die Renderskalierung auf einen Wert größer 100 Prozent stellen und im Grafikmenü bei der Kantenglättung die höchste Stufe auswählen.**

Full HD, TAA, dyn. SSAA (ca. 140-160 %)

**Schließlich müssen Sie noch ihre gewünschte Framerate bei „Adaptiver Qualität“ festlegen. Wenn Ihre Leistung ausreicht, appliziert Odyssey nun Supersampling, die Qualität reicht von der gewählten Renderskalierung (Beispiel: 150 % entspricht 1,5 × 1,5 OGSSAA) als Maximum bis zu der ausgewählten Auflösung (also etwa Full HD) hinab.**

**Das beste Resultat erzielen Sie mit einem zusätzlichen Framelock, welches Ihrer Hardware noch etwas Spielraum bietet.**

unsauber und zugleich unscharf wirken.

Aktivieren wir DLSS, so wird die Grafik statt mit 3.840 × 2.160 Pixeln mit 2.560 × 1.440 Bildpunkten berechnet. DLSS bereitet das Bild dann mit Hilfe hochauflösenden Materials von Nvidias Serverfarmen intelligent und unterstützt von einer temporalen Kantenglättung auf. Das mit DLSS aufgewertete WQHD-Bild wirkt wesentlich sauberer und ruhiger als selbst die Präsentation in Ultra HD. Die Kanten sind sehr glatt, die grieselige Unschärfe des standardmäßig in Final Fantasy 15 eingesetzten TAAs reduziert sich, die pixeligen Haare der Figuren und Protagonisten sowie die Vegetation werden wirksam geglättet.

## Veränderte Optik

Final Fantasy 15 ist mit DLSS für das Auge wesentlich angenehmer und weniger anstrengend. Allerdings fallen auch sofort deutliche optische Unterschiede auf, denn vollständig originalgetreu hinterlässt das KI-Upscaling die Optik nicht.

Effekte wie Schatten, Screen-Space-Reflections und die Umgebungsverdeckung verändern ihre Struktur und Dichte, wohl der niedrigeren internen Auflösung geschuldet. Bei einigen Reflexionen und Verschattungen treten mit DLSS auch kleinere Fehldarstellungen auf, die mit regulärer, nativ berechneter Auflösung weder in WQHD noch UHD zu beobachten sind.

## Mit Pinsel und Ölfarbe

Doch die größten Unterschiede zeigen sich bei der Vegetationsdarstellung und anderen feinen Elementen wie Haaren. Falls Sie sich schon einmal mit Pixel-Art-Shadern beschäftigt haben, der Look dieser feinen Elemente erinnert entfernt an Scaler wie SAI2x oder SuperSAI. Etwas abstrakter formuliert: Haare, Vegetation und einige andere Elemente wirken bisweilen ein wenig wie mit einem Pinsel und mit Ölfarbe gemalt.

Das ist per se nichts Schlechtes, der Look passt zudem zu Final Fantasy, zudem erhöht sich mit DLSS die Leistung gegenüber Ultra HD mit TAA und trotzdem wirkt das Bild sauberer – aber es ist eben nicht das gleiche Ergebnis. DLSS ist in Final Fantasy 15 stilverändernd, wenn auch nur geringfügig.

## Nachteil: Flimmern und Ghosting

Es gibt außerdem neben den Effekten einige weitere Elemente, welche die gegenüber UHD niedrigere Renderauflösung dokumentieren. Einige feine Strukturen wie Zäune oder die Spalten in Holzbrücken flimmern mit Deep-Learning-Supersampling stärker als in Ultra-HD mit TAA. Außerdem kann mit DLSS in einigen Situationen ein starkes Ghosting provoziert werden.

Dennoch würden wir die hochskalierte und KI-verbesserte WQHD-Präsentation dem UHD-Original wohl vorziehen, der höheren Sauberkeit, geringerer TAA-Unschärfe und besserer Performance sei Dank.

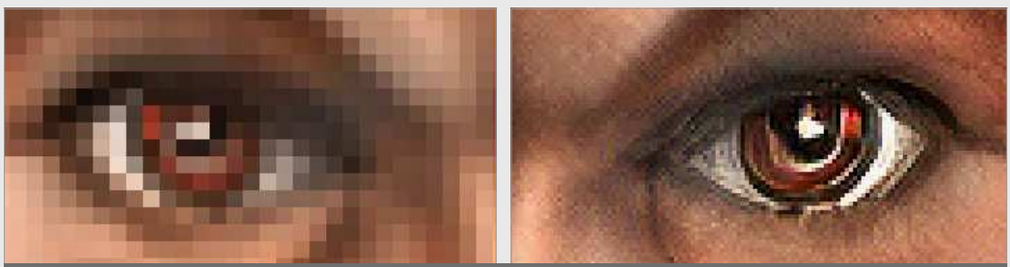
## Wie ESGRAN ältere Spiele aufhübschen kann

**Nicht nur Nvidia setzt mit DLSS und seinen Geforce-RTX-Grafikkarten auf KI, sondern auch Modder wollen auf diese Weise ältere, aber immer noch beliebte Spiele visuell verschönern**

**Ein beeindruckendes Beispiel ist dem Modder DassiD gelungen, der eine Technik namens Enhanced Super-Resolution Generative Adversarial Networks (ESRGAN) verwendet. Damit wurden die Texturen von The Elder Scrolls: Morrowind durch KI-Upscaling verbessert, ohne dass dabei störende Artefakte entstehen. Das Resultat sind Texturen mit der vierfachen Auflösung der Originalversionen (siehe Gesichts-Bilder unten).**

**Das Prinzip basiert auf zwei neuronalen Netzen. Der etwas vereinfachte Ansatz funktioniert wie folgt: Von den vorhandenen Grafiken werden niedrig aufgelöste Varianten erstellt. Das erste neuronale Netz erzeugt nun hochauflösende Fälschungen dieser pixeligen Ausgangsgrafiken. Das zweite neuronale Netz urteilt anschließend, wie gut das errechnete Bild zum Originalmaterial passt. Ist das Ergebnis zufriedenstellend, kann es auch auf das Originalmaterial selbst angewendet werden.**

Kein Treppeneffekt mehr beim Dach oder Baum. Die filigranen Strukturen, die beim Upscaling sonst verloren gehen, wurden durch die KI erhalten



ESRGAN zeigt, zu was es in der Lage ist: Die filigranen Strukturen im Auge rechts sind so nicht im Originalbild enthalten und wurden überzeugend hinzuhalluziniert

# Kantenglättung

Rainbow Six Siege bietet gutes Anschauungsmaterial, wenn es um Kantenglättungen geht. Zum einen unterstützt der Titel mehrere technische Herangehensweisen, darunter neben FXAA auch Nvidias TXAA, MSAA, sowie eine weitere temporale Kantenglättung. Zum anderen ist die Optik des Taktik-Shooters eher zweckdienlich gehalten: Weder blenden überbordernde Post-Processing-Effekte, noch lenken spektakuläre Objekte und Details das Auge ab. Dieses Bild wurde in Ultra HD geschossen. Trotz der hohen Auflösung sind Pixeltreppen klar zu identifizieren. Auf einem großen Display und in Bewegung ist das Geflimmer selbst in Ultra HD sehr lästig und insbesondere bei hohen Kontrasten kaum zu übersehen (1), (4). Das Zuschalten von FXAA hilft nur wenig und kommt mit zusätzlicher Unschärfe daher – insbesondere filigrane Strukturen wie die Vegetation verlieren Details, behalten in Bewegung aber gleichzeitig eine hohe Unruhe bei (2). Das langsam aussterbende Multisampling demonstriert mit schlechter Effektivität sowie hohen Performance-Kosten und damit einer eher miserablen Effizienz, weshalb MSAA in aktuellen Titeln kaum noch zum Einsatz kommt (3). Das auf MSAA basierende TXAA ist ein wenig schärfer als das TAA, kostet allerdings viel Leistung und bietet somit recht wenig Anreiz, es dem TAA vorzuziehen.

Kein Antialiasing

FXAA (hoch)

4x MSAA

TAA

4x TXAA

# Moderne Grafikeffekte: Tiefenschärfe

Die Tiefenschärfe sorgt für Unschärfe, aber auch für mehr Tiefe und hilft beim Herausstellen von Bildinhalten. Und sie ist außerdem ein visueller Trick, um Blick und Aufmerksamkeit des Betrachters zu steuern.

## Unscharfer Hintergrund

Die Tiefenschärfe simuliert die sogenannte Akkomodation, also die dynamische Brechkraft der Linse des menschlichen Augens. Dies kann man leicht selbst überprüfen: Nimmt man einen Finger, hält diesen etwa 30 bis 50 cm vor seine Augen und fokussiert ihn, wird dahinter Liegendes unscharf. Nimmt man ein Objekt im Hintergrund ins Visier, wird wiederum der Finger undeutlich. Das liegt daran, dass wir im dreidimensionalen Raum sehen und sich die Brechkraft der Linsen unserer Augen ändert, je nachdem, wie weit das fokussierte Objekt entfernt ist.

Vor dem Fokus sind wir weit-, dahinter kurzsichtig. Schauen wir dagegen auf einen Bildschirm, eine Kinoleinwand oder ein Foto und versuchen, in einem dortigen Bild den gleichen Effekt herbeizuführen, so funktioniert das nicht. Das ist durchaus logisch, schließlich blicken wir auf eine zweidimensionale Fläche. Ähnliches gilt übrigens selbst für VR-Brillen - dort schauen wir mit jedem Auge auf eine Fläche. Das gaukelt uns zwar Dreidimensionalität vor, doch ist echtes räumliches Sehen und Fokussieren auch in VR nicht möglich.

Schauen Sie sich einmal den Screenshot auf der rechten Seite an: Rainbow Six Siege nutzt hier keine Tiefenschärfe. Setzen Sie ihren Fokus zum Beispiel einmal auf die Kimme des Gewehrs. Wird das dahinterliegende Polizeiauto unscharf? Nein, wird es nicht. Aber das sollte es, denn normalerweise würde sich unser Auge auf diesen Punkt einstellen, Dahinterliegendes würden wir wahrnehmen, als wären wir kurzsichtig.

Auch Elemente vor dem Fokuspunkt wären unscharf, dort wären wir weitsichtig. Bei diesem Umstand lässt sich unser Sehen also nicht austricksen: Gleich ob Sie das Bild nun im Druck, auf Ihrem Monitor sehen oder gar durch eine VR-Brille betrachten würden: Wir schauen immer auf eine zweidimensionale Fläche (bei VR sind es deren zwei) und das Auge stellt sich auf diese Fläche scharf. Nehmen wir aber etwa unsere Hand und platzieren sie zwischen Auge und Display und fokussieren den Blick auf Hand oder Monitor, wird das jeweils andere Objekt unscharf. Das wäre reale Tiefenschärfe und sie funktioniert nur im dreidimensionalen Raum.

## Dynamische Unschärfe

Depth-of-Field in Spielen greift diese Problematik auf: Es simuliert den Umstand, dass wir nur eine bestimmte „Ebene“ in einem dreidimensionalen Raum scharf wahrnehmen können. Doch es gibt dabei ein großes Problem: Das Spiel weiß nicht, wohin unsere Augen blicken, und auch nicht, welchen Punkt im dreidimensionalen, virtuellen Raum wir betrachten und damit scharf sehen möchten.

Ein Problem bei einer adaptiven Tiefenschärfe, welche in einigen Spielen und vielen ENB-Konfigurationen zum Einsatz kommt: Nur der Fokuspunkt (meist um das Faden-

Rainbow Six Siege verzichtet – wie viele andere Spiele – auf eine Tiefenunschärfe; alle Elemente sind immer scharf zu erkennen, egal wie weit sie vom Protagonisten entfernt sind

Mass Effect 3 nutzt den Tiefenschärfe-Effekt geschickt, um das Auge des Spielers in Zwischensequenzen zu dirigieren

kreuz) ist scharf, das Depth-of-Field indes nur dann überzeugend, wenn der Spieler auf dem Display genau diesen Punkt fokussiert. Entfernt sich der Blick von ihm, funktioniert die Illusion nicht mehr, das Ergebnis kann befremdlich wirken oder gar leichtes Unwohlsein auslösen. Eine adaptive Tiefenschärfe wird daher nur relativ selten eingesetzt oder wird, wie beispielsweise in Skyrim SE, in der Funktion stark eingeschränkt. Titanfall 2 hingegen ist eines der (wenigen) Beispiele für eine ambitioniert umgesetzte adaptive Tiefenschärfe.

## Tricks aus Hollywood

Viel häufiger orientieren sich Spiele-Entwickler jedoch an Film und Foto. Dort wird eine andere Besonderheit des menschlichen Sehens ausgenutzt: Beim Betrachten eines Bildes neigt das Auge dazu, Unscharfes zu ignorieren und scharfe Inhalte zu fokussieren. Testen Sie das ruhig einmal selbst auf der rechten Seite. Ihr Blick und Ihre Aufmerksamkeit wandern automatisch dorthin, wo wir (erstes Bild) oder die Entwickler ihn haben wollen.

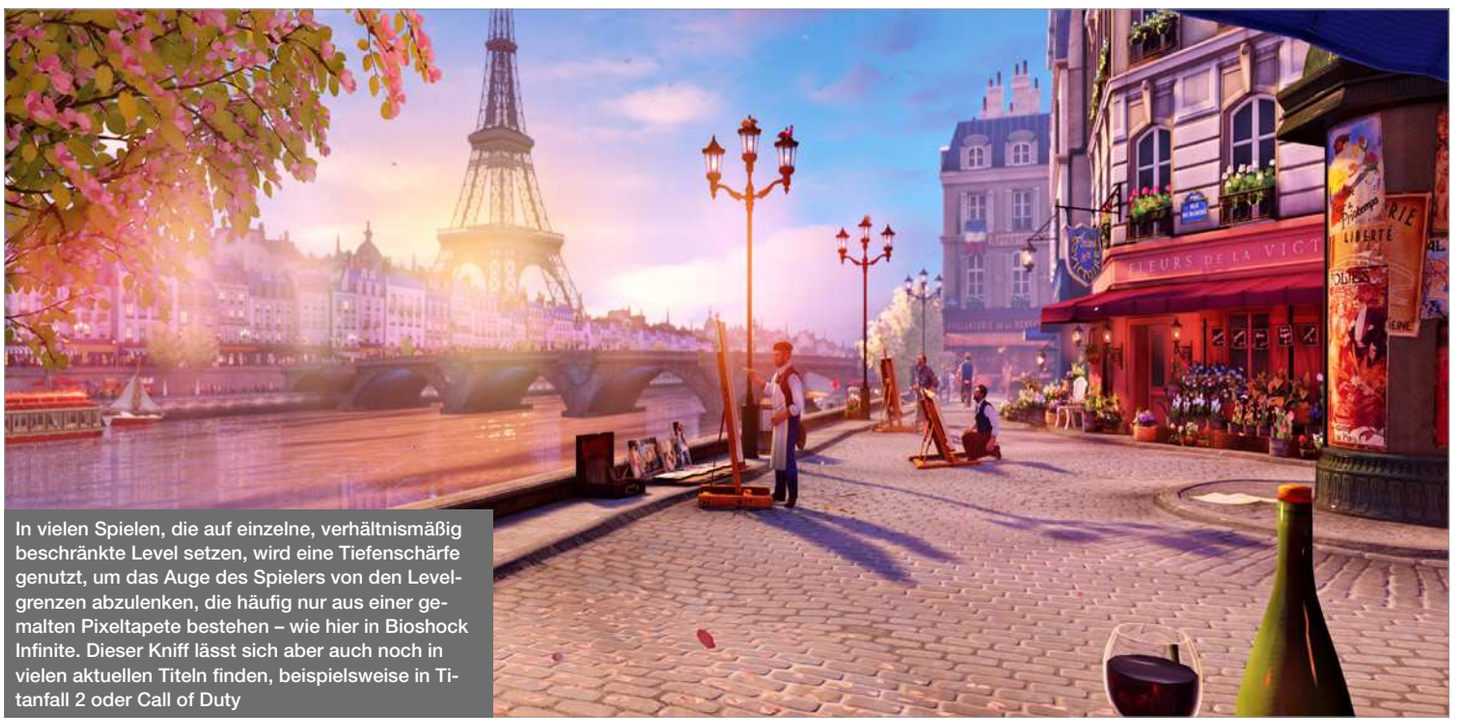
Dieser Trick, den Blick zu steuern, wird häufig in Zwischensequenzen oder bei Gesprächen mit NPCs eingesetzt, um die Aufmerksamkeit des Spielers zu lenken. Das funktioniert auch andersherum: Wird etwas unscharf gezeichnet, wird das Auge des Betrachters von diesem Element abgelenkt und ist wenig geneigt, dort viel Aufmerksamkeit zu investieren. Damit eignet sich eine solche Unschärfe, kritische, detailarme Elemente oder Tricksereien zu verdecken. Häufig ist dies in „schlauchigen“ Shootern oder Spielen anzufinden, die auf einzelne, beschränkte Level setzen, um Begrenzungen oder den niedrigen Detailgrad zu kaschieren. Open-World-Titel begrenzen die Sichtweite eher selten auf diese Art.

## Aktuelle Beispiele

Beispiele für eine gelungene Umsetzung liefert dagegen unter anderem Mass Effect Andromeda: Die Tiefenschärfe ist maximiert eine echte Augenweide. Sie kommt in Zwischensequenzen und Gesprächen zum Einsatz und wird analog zu Film und Fernsehen dazu genutzt, das Auge des Zuschauers zu leiten. Figuren, die gerade eine Funktion erfüllen oder das Gespräch führen, werden in den Vordergrund gestellt, der häufig chaotische Hintergrund der Szenen dagegen unscharf dargestellt. Dies erleichtert es dem Betrachter, der Szene zu folgen und ist für das Auge außerdem weniger anstrengend. Die sehr hohe Qualität der Tiefenschärfe sowie die zusätzlichen Bokeh- und Chroma-Effekte kosten indes auch viel Leistung: Die Cutscenes gehören zu den anspruchsvollsten Szenen.

In The Witcher 3 ist die Tiefenschärfe in Gesprächen eher unauffällig

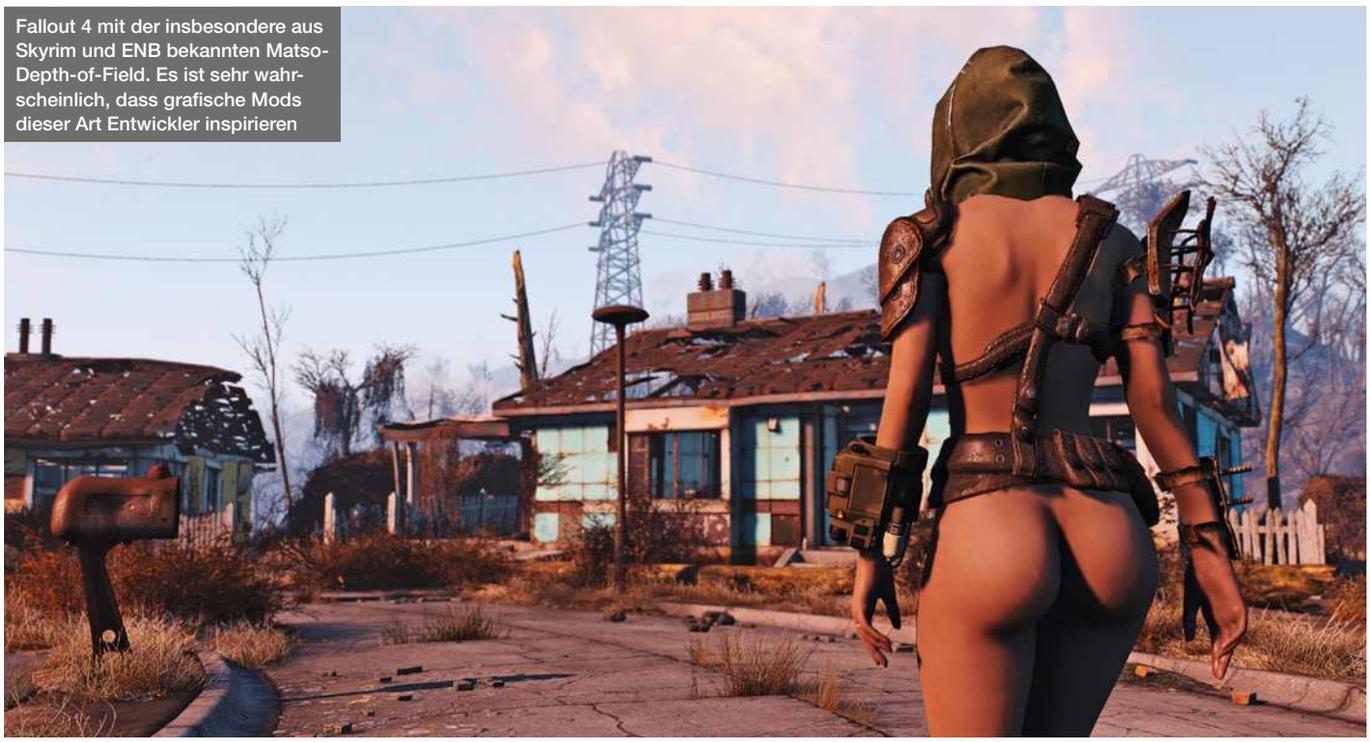
Vor einigen Jahren wurde eine anspruchslose und „unechte“ Tiefenschärfe (meist ein simpler, niedrig aufgelöster Gauss-Filter) auch dazu genutzt, um dahinter fehlende Details oder Begrenzungen zu verstecken.



In vielen Spielen, die auf einzelne, verhältnismäßig beschränkte Level setzen, wird eine Tiefenschärfe genutzt, um das Auge des Spielers von den Levelgrenzen abzulenken, die häufig nur aus einer gemalten Pixeltapete bestehen – wie hier in Bioshock Infinite. Dieser Kniff lässt sich aber auch noch in vielen aktuellen Titeln finden, beispielsweise in Titanfall 2 oder Call of Duty

# Tiefenschärfe

Beispiele für den Einsatz einer Tiefenschärfe: Im ersten Bild demonstrieren wir die Funktionsweise einer händisch abgestimmten Depth-of-Field, wie es beispielsweise in einer Zwischensequenz Verwendung finden könnte: Der Blick des Betrachters wird von der unscharf dargestellten Umgebung auf die in den Fordergrund gestellten Spielfigur gelenkt. Eventuell ist daran auch noch ein anderer Faktor beteiligt, jedoch ist das nicht der Punkt. Der nackte Hintern ist allerdings thematisch interessant, da wir hier eine Art Grafikmodifikation reproduzieren, die bei der Verbreitung der Bokeh-Depth-of-Field eine gewisse Rolle gespielt haben dürfte: Es gibt seit Jahren unzählige ENB-Modifikationen für Skyrim mit diesem spezifischen, oder einem sehr ähnlichen Tiefenschärfe-Effekt. Millionen Bilder fluten das Netz. Es ist wohl kaum strittig, dass sich Entwickler von solcherlei Modding-Trends beeinflussen lassen und Spiele entsprechend aufrüsten. So wurde die Skyrim Special Edition mit einer adaptiven Depth-of-Field ausgestattet – ein offenkundiger Wink Richtung Modding-Community. Eine adaptive Tiefenschärfe hat gegenüber einem händisch platzierten Fokus aber einige Nachteile: Da der Spieler den Fokus selbst mit seinem virtuellen Blick setzt, also beispielsweise mit dem Fadenkreuz, muss er auch aktiv selbst auf den entsprechenden Punkt auf dem Monitor schauen, ansonsten wirkt die Tiefenschärfe eher befremdlich.



Fallout 4 mit der insbesondere aus Skyrim und ENB bekannten Matso-Depth-of-Field. Es ist sehr wahrscheinlich, dass grafische Mods dieser Art Entwickler inspirieren

Die neu hinzugekommene adaptive Tiefenschärfe in der Skyrim Special Edition

Die adaptive Depth-of-Field beim Zielen in Titanfall 2

# Moderne Grafikeffekte: Globale Beleuchtung

Die Enlighten-Middleware gehört wohl zu den aktuell verbreitetsten Techniken. Neben der Unity-Engine nutzte schon Battlefield 3 diese Art Globale Beleuchtung

Assassin's Creed Unity war einer der ersten Titel, welcher die Fähigkeiten der „Next-Gen"-Konsolen ausschöpfte und diese direkt leistungstechnisch überforderte. Neben Physically Based Rendering kommt eine hübsche Globale Beleuchtung zum Einsatz, die allerdings vorberechnet ist. Die Beleuchtungsdaten machen 16,8 von 40,4 GiByte aus, beinahe die Hälfte der gesamten Spielgröße (ohne DLCs) – was die Komplexität der Technik verdeutlicht

Eine Globale Beleuchtung („Global Illumination") sorgt für glaubhaftes und dynamisches Licht in modernen Spielen.

## Globale Beleuchtung

Die Globale Beleuchtung ist in den vergangenen Jahren immer wichtiger geworden. Doch was kann man sich eigentlich unter einer „globalen" Beleuchtung vorstellen?

Wenn in der Realität ein Lichtstrahl auf eine Oberfläche trifft, wird ein Teil absorbiert, der Rest wird reflektiert. Dieses abgestrahlte Licht bestimmt, mit welcher Helligkeit und Färbung wir eine Oberfläche wahrnehmen. Der reflektierte Strahl trifft nun indirekt auf weitere Oberflächen, das Spiel wiederholt sich.

Da nun jedoch nur noch ein Teil des ehemals vorhandenen Spektrums dort eintrifft, findet neben einer indirekten Beleuchtung auch ein Farbtransfer statt: War die erste Oberfläche blau (sie wirft also den blauen Anteil des sichtbaren Lichtspektrums zurück), wird die zweite Oberfläche blau eingefärbt – vorausgesetzt, sie schluckt das Licht nicht (wenn sie beispielsweise schwarz wäre).

In der Realität geschieht dies so häufig, bis der ausgesandte Lichtstrahl komplett absorbiert wurde.

## Die „Realität" in Spielen

Sie ahnen es vielleicht schon: Für heutige Hardware ist eine solch komplexe Berechnung in Spielen nicht zu stemmen, da natürlich nicht nur die Reflexionen und Farbübertragungen eines einzelnen Strahls berechnet werden müssen, sondern die unzähliger weiterer Strahlen. Trotzdem kann nur noch schwerlich auf eine Globale Beleuchtung verzichtet werden. Gründe dafür sind die gesteigerte Dynamik und Komplexität moderner Spiele, insbesondere bei Open-World-Titeln samt wechselnder Witterungsverhältnisse und Tageszeiten.

Ein anderer Grund ist der höhere Realismusbezug durch Phyically Based Rendering: Da die Materialien physikalische Attribute besitzen, die bei Lichteinfall die Realität imitieren, muss logischerweise auch die Beleuchtung möglichst authentisch ausfallen, um diese Eigenschaften zu bedienen. Ansonsten wirkt das Bild nur bedingt glaubhaft.

Daher kommt in beinahe jedem aktuellen Titel eine Globale Beleuchtung ins Spiel. Selbst die Indie-Engine Unity nutzt in Form von Enlighten eine solche (siehe Bild links oben). Die Präzision und Dynamik dieser Techniken sind natürlich eingeschränkt und werden zusätzlich durch andere Rendertricks ergänzt, etwa einer Umgebungsverdeckung oder Screen-Space-Reflections.

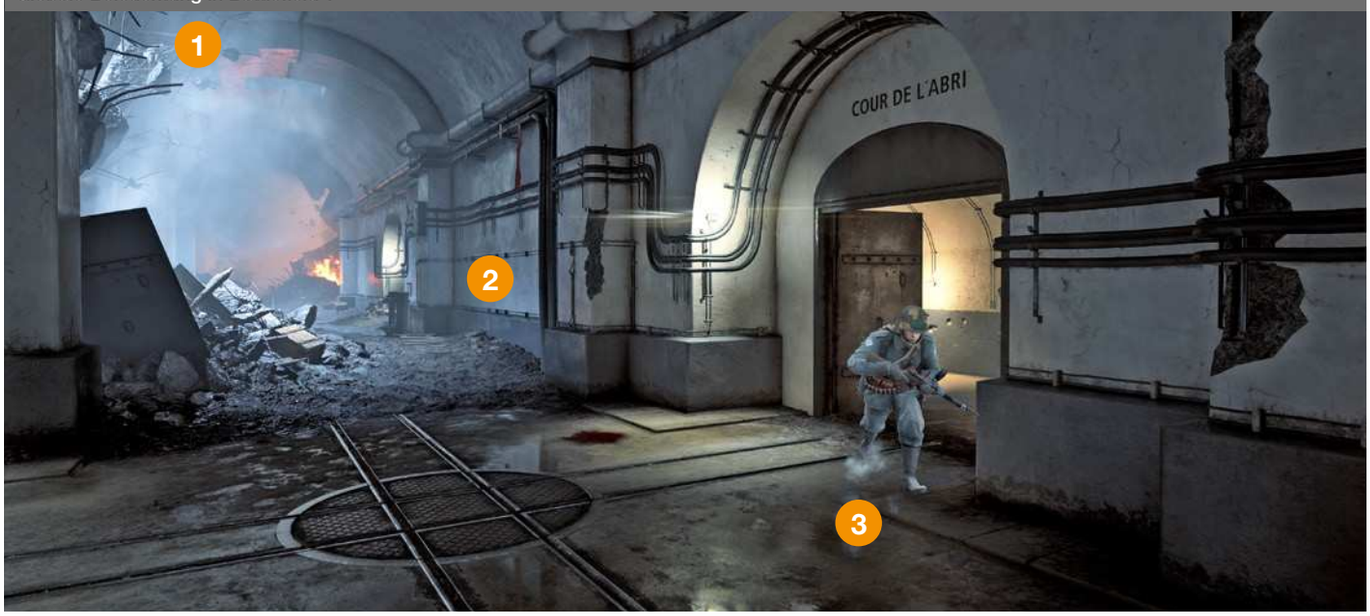
## Voxel Ambient Occlusion

Grundsätzlich würde eine vollumfängliche, detaillierte Globale Beleuchtung einige Effekte überflüssig machen, so ist beispielsweise Nvidias VXAO („Voxel Ambient Occlusion") ein Teil von VXGI („Voxel Global Illumination"). Doch noch ist die Technik nicht ausgereift genug beziehungsweise eine solche GI zu teuer, daher muss abstrahiert werden. Dabei nutzen Entwickler verschiedene Ansätze, von denen wir einige auf den folgenden Seiten vorstellen.

# (Halb-)dynamische Globale Beleuchtung

Die (halb-)dynamische Globale Beleuchtung via Enlighten-Middleware ist weit verbreitet und unter anderem in der Unreal Engine 4 integriert. Aber natürlich kann die Middleware in Spielen unterschiedlich genutzt und auch in Sachen Qualität variiert werden. Enlighten verwendet einen dynamischen Ansatz für Levelarchitektur, was insbesondere bei Titeln wie Battlefield 1 mit zerstörbaren Umgebungen offenkundige Vorteile bringt. Die Licht- und Farbübertragungen sind prinzipiell sehr überzeugend, wobei der Effekt beispielsweise im ebenfalls mit Enlighten ausgestatteten Mirror's Edge Catalyst eventuell etwas zu akzentuiert ausfällt. Die GI wird mit Screen-Space-Reflections (1) sowie einer Umgebungsverdeckung ergänzt (2), obwohl die Globale Beleuchtung die gleichen Effekte schon in abgemilderter Form mit sich bringt. Die Figuren werden per Light Probes beleuchtet. Dies sind im Level verteilte unsichtbare „Sonden", die umliegende Beleuchtungsdaten sammeln und bei Kontakt an Spielfiguren und Objekte abgeben. Der Soldat bei (3) kommt von draußen, hat noch keine Light Probe im Inneren getroffen und trägt daher noch die optisch inkorrekte Beleuchtung von Außen.

Globale Beleuchtung in Battlefield 1




Gut erkennbare, jedoch auch etwas unpräzise und stark akzentuierte Farbübertragung in Mirror's Edge Catalyst

# Voxelbasierte Globale Beleuchtung

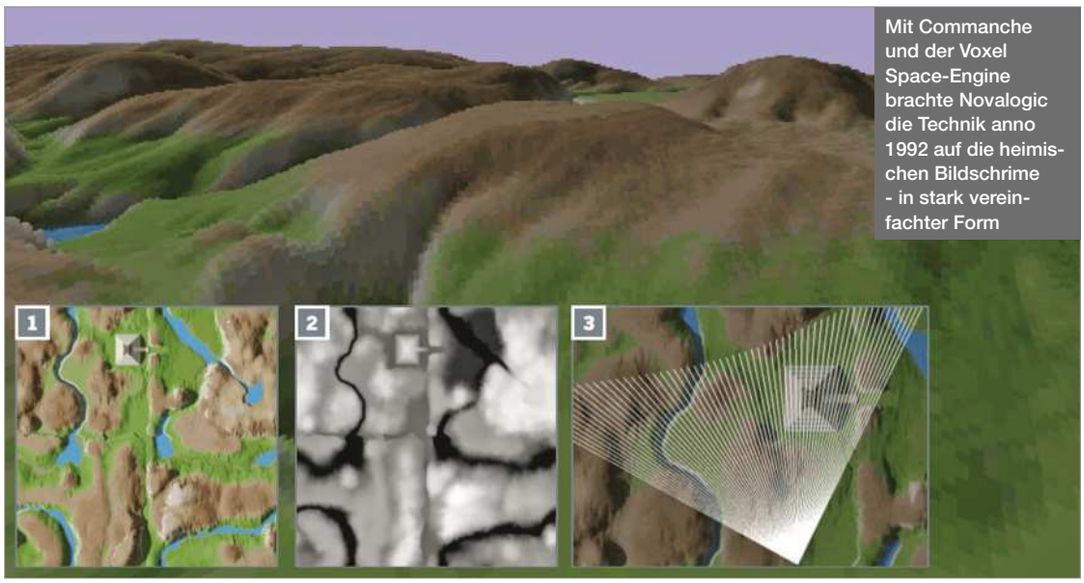
Mit Voxeln kann eine komplexe Szene abstrahiert und vereinfacht werden – auch intern, etwa zur Berechnung einer Globalen Beleuchtung. Durch diese Reduktion kann wiederum die eigentliche Berechnung der GI deutlich komplexer ausfallen und dem Raytracing weiter angenähert werden. Eine solche Herangehensweise demonstriert etwa die Lunar Lander Demo samt Nvidias VXGI. Dank der sehr komplexen und realistischen Ausleuchtung der Szene sind keine „künstlichen" Ambient Lights und ähnliche Tricks nötig. Allerdings ist VXGI noch dermaßen anspruchsvoll, dass selbst eine stark übertaktete GTX 1080 Ti schon in Full HD an ihre Grenzen stößt und deutlich unter 60 Fps absacken kann – und das bei einer im Vergleich zu Spielen eher simplen Szene. Einen ähnlichen Ansatz verfolgt Crytek mit SVOTI, welches mit diversen Kniffen arbeitet, darunter ein LoD und Berechnungen über mehrere Frames. SVOTI zeigt temporale Nebeneffekte, wirkt in einigen Belangen nur grobflächig und nutzt zur Farbübertragung die Diffuse-Texturen, nicht die tatsächliche Färbung in der vorliegenden Szene, was zu einigen kleineren Unstimmigkeiten führt. Insgesamt ist die Vorstellung der sich bislang noch in Entwicklung befindlichen Technik in den ebenfalls noch nicht vollendeten Spielen jedoch sehr eindrücklich – insbesondere unter Berücksichtigung der eher geringen Leistungskosten.

Nvidias voxelbasierte Globale Beleuchtung VXGI ist schick und präzise, allerdings auch sehr kostspielig und Nvidia-exklusiv

Die voxelbasierte Globale Beleuchtung SVOTI von Crytek ist zur Zeit (noch) ein wenig eingeschränkt, aber auch sehr schick und verhältnismäßig günstig

# Moderne Grafikeffekte: Volumetrische Pixel (Voxel)

Aktuell sind Voxel wieder in aller Munde. Den hippen Status verdankt die Technik insbesondere Indie-Titeln wie Minecraft, 7 Days to Die, Nex Machina und anderen Spielen in „Klötzchen-Optik“, wie Eldritch. Dabei ist Voxel-Grafik keine neue Erfindung und tatsächliche Voxel bekommen Sie in keinem der genannten Titel zu Gesicht. Was Sie aktuell in Spielen mit Voxel-Technik sehen, ist Voxel-basierte Polygongrafik. Die Voxel dienen der Engine lediglich als Basis, um mit diesen Geometrie oder Effekte zu erzeugen.




Mit Commanche und der Voxel Space-Engine brachte Novalogic die Technik anno 1992 auf die heimischen Bildschrime - in stark vereinfachter Form

Bild: simulationcorner.net

## Voxel-Technik

Dabei waren und sind Voxel eine interessante sowie bereits seit langem bekannte Technik; vor dem Siegesmarsch der 3D-Beschleunigerkarten träumte so mancher gar davon, dass Voxel, nicht Polygone, die Grafik der Zukunft ausmachen würden. Ursprünglich stammt die Technik aus der Medizin, genauer der Magnetresonanztomographie. Anno 1992 schwappte Voxel-Grafik dann in Form der Voxel Space Engine auf den PC herüber.

Die von Novalogic patentierte Technik kam unter anderem in der Hubschraubersimulation Comanche zum Einsatz und war damals das Nonplusultra in Sachen Landschaftsdarstellung. Die Engine zeichnete Landschaften weitaus glaubwürdiger als damalige 3D-Engines mittels Polygonen. Sogar die US Army nutzte Simulationen auf Basis der Novalogic-Engine. Auch andere Spiele experimentierten mit Voxel-Grafik, als wohl fortschrittlichstes Spiel mit Voxelgrafik gilt bis heute Outcast von 1999.

## Pixel vs. Voxel

Während der Begriff Pixel aus „Picture“ (Pix) und „Element“ (el) zusammengesetzt wird, besitzt ein Voxel einen Körper, daher „Volume“ und „Element“ – Voxel. Etwas konkreter ist die ebenfalls gebräuchliche Herleitung aus „Volume“ und „Pixel“: Ein Voxel ist demnach ein Bildpunkt mit Volumen, ein „3D-Pixel“ also. Dieses hat im Vergleich zum regulären Pixel drei statt zwei Koordinaten (x, y, z) sowie einen einzelnen Farbwert.

## Voxel-Grafiken

Um es ein wenig bildhafter zu beschreiben: Man kann sich einen Voxel wie einen Legostein vorstellen. Aus diesen lässt sich ein dreidimensionales Gebilde bauen. Je mehr Bausteine man nutzt, desto komplexer und feiner fällt ein solches Objekt aus. Aus Voxeln lässt sich somit grundsätzlich eine hochdetaillierte, dynamische sowie ausgesprochen organische Grafik realisieren. Doch natürlich gibt es einen Haken: Die Darstellung einer hochauflösenden Voxel-Szene wäre extrem rechenaufwendig und die auf Beschleunigung von Polygonen ausgelegte Grafikkarte nur sehr begrenzt hilfreich – die Voxel müssten zumindest zu weiten Teilen von der CPU gehandhabt werden.

Außerdem müsste in einem dreidimensionalen Gitter für jeden (potenziellen) Voxel ein Speicherplatz reserviert werden, bei dynamischer Darstellung gar mit ständigem Lese- und Schreibzugriff. Je höher aufgelöst und größer eine Voxel-Welt ausfällt, desto mehr Speicher benötigt sie. Der Speicher stellt damit neben der leistungsintensiven Darstellung die wohl größte Problemstelle bei Voxeln dar. Trotz Optimierungen beziehungsweise Speichereinsparungen via (Sparse-)Octree-Kompression bekommen Sie „echte“ Voxel-Grafik in Spielen daher nur selten zu Gesicht

Voxel haben ihre Ursprünge in wissenschaftlichen 3D-Scans, genauer: der Magnetresonanztomographie (MRI)

Bild: Wikipedia Commons

# Voxel-Grafik

Mehrere Jahre war nicht gewiss, ob Spiele der Zukunft mit Polygonen oder Voxeln dargestellt werden würden. Doch dies sollte sich um die Jahrtausendwende klären: Mit der Verbreitung von Beschleuniger-Karten und moderner(er) GPUs hatten Voxel ausgedient. Entwickler setzten nun fast ausschließlich auf Polygone für die 3D-Darstellung. Die Anzahl nutzbarer Vielecke stieg zu dieser Zeit massiv an, zugleich wuchsen die Fähigkeiten der Grafikkarten. Das Jahr 1999 ist gleich mehrfach interessant, denn zum einen erschien mit Nvidias Geforce 256 eine Grafikkarte, die mittels Hardware Transform & Lighting den Grundstein für die moderne, GPU-beschleunigte Geometrie- und Shader-Grafik legte.

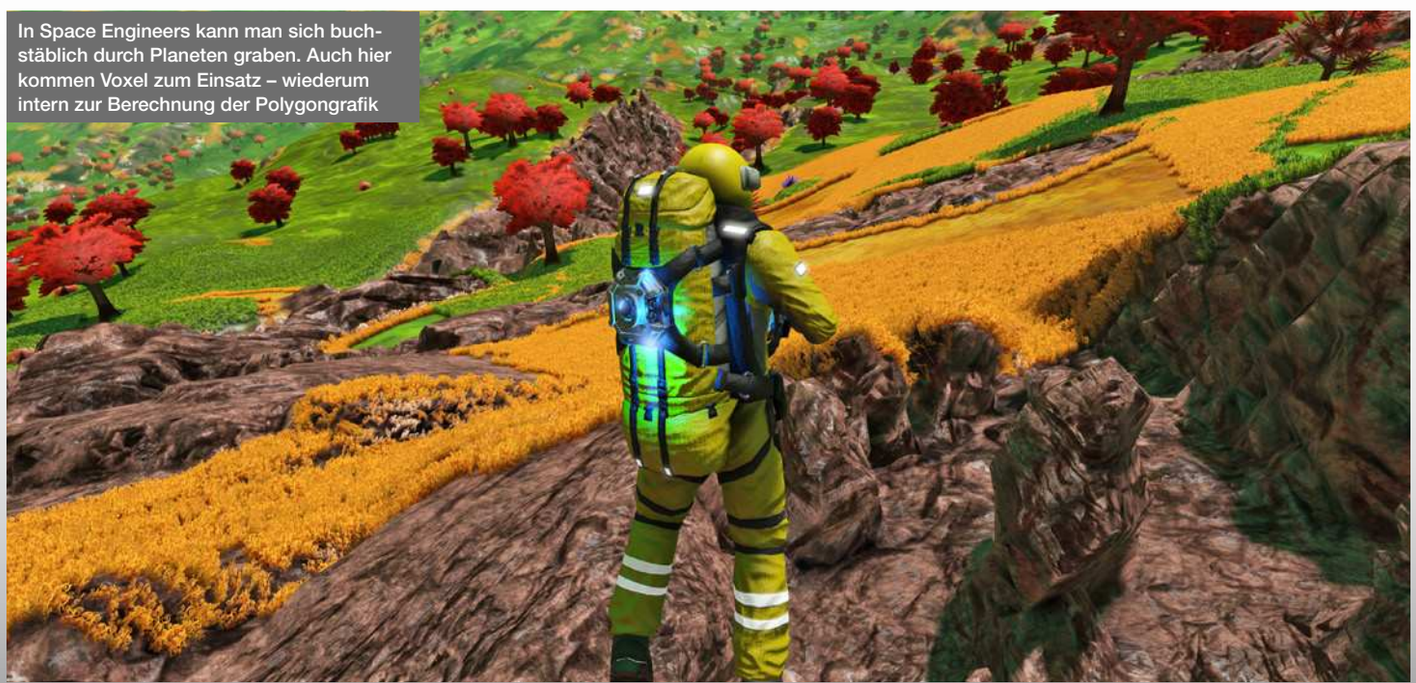
Zum anderen erschien im selben Jahr mit Outcast eines der letzten und bis dato wohl aufwendigsten Spiele mit Voxel-Grafik. Die zum Einsatz kommende Technik nutzt – ähnlich der Voxel Space Engine – Höhen-Maps und Raycasting zur Terrain-Darstellung, Objekte und Figuren werden mittels Polygonen dargestellt. Outcast war aufgrund der aufwendigen Voxel-Darstellung ein massiver Prozessor-Killer und bringt auch heute, trotz Mehrkernoptimierungen, selbst moderne Achtkern-CPUs ins Schwitzen. Ein weiterer Voxel-Titel aus dem Jahre 1999 ist Command & Conquer Tiberian Sun. Das Strategiespiel nutzt damals übliche 2D-Kacheln für die Landschaftsdarstellung, Einheiten wie Fahrzeuge bestehen dagegen aus Voxeln.

Outcast ist das wohl fortschrittlichste Spiel mit Voxel-Grafik. Der Ansatz ähnelt der Raycasting-Technik von Novalogic, die Landschaft wird aus Texturen generiert

Command & Conquer Tiberian Sun nutzt für die Darstellung der Einheiten Voxel. Diese können sich dank der Technik stufenlos drehen und mit dem Untergrund agieren

Wenn heutzutage in Spielen Voxel zum Einsatz kommen, dann ist dies in praktisch allen Fällen nicht auf den ersten Blick ersichtlich. Ja, einige „blockige“ Titel wie Minecraft nutzen ein Voxel-Raster und Voxel zur Berechnung der einzelnen Quader, aus denen das Spiel besteht. Doch handelt es sich bei den sichtbaren Elementen um Polygonkörper samt üblicher Geometrie- und Shaderberechnungen. Die Voxel dienen lediglich als Basis für die Polygondarstellung. Trotzdem birgt die Technik einige ihrer Vor- und Nachteile: Zu den positiven Aspekten zählt die Dynamik – Elemente können erstellt und zerstört werden. Das ist sehr vorteilhaft in Sandbox-Titeln, in denen Spieler starken Einfluss auf die Umgebung ausüben und beispielsweise Ressourcen (optisch) abbauen können. Zudem ist es verhältnismäßig einfach, mit Voxeln glaubhafte, prozeduale Inhalte zu generieren. Auf der anderen Seite benötigen Voxel noch immer viel Rechenzeit, außerdem ist der Speicherhunger trotz Octree-Kompression (nur auf grobmöglichste, benötigte Daten wird zugegriffen) problematisch – auch das kennen Sie eventuell aus Minecraft: Große, komplexe Maps lassen die Prozessor- und Speicherlast aufgrund der hohen Voxel-Dichte in die Höhe schnellen. Dennoch sind intern berechnete Voxel bestens geeignet, um aus groben, abstrakten Strukturen detaillierte Polygongebilde zu erzeugen. Voxel werden außerdem genutzt, um aufwendige volumetrische Effekte zu realisieren (z. B. SVOTI, VXAO).

No Man‘s Sky nutzt Voxel intern, um prozeduale Inhalte zu generieren und komplexe Polygonstrukturen für die sichtbare 3D-Grafik zu erstellen



In Space Engineers kann man sich buchstäblich durch Planeten graben. Auch hier kommen Voxel zum Einsatz – wiederum intern zur Berechnung der Polygongrafik

# Moderne Grafikeffekte: Schatten

Schatten wirkten früher oft klobig und pixelig. Dank moderner Verfahren sind diese Zeiten glücklicherweise vorbei. Welche Schatten-Technik macht in Zukunft das Rennen?

Eine hochauflösende, gefilterte und saubere Shadow Map in Ghost Recon: Wildlands

Ohne zumindest halbwegs glaubhafte Schatten kommt heute kaum ein Spiel mehr aus. Sie ergänzen die seit Jahren komplexer werdenden Beleuchtungstechniken, sorgen für Tiefe und Kontrast und lassen virtuelle Welten glaubhafter wirken.

## Schatten

Doom 3 setzte 2004 so etwas wie den Anfangspunkt der modernen Schattendarstellung. Entwicklerlegende John Carmack erdachte für den Kult-Shooter die so genannten Stencil-Shadows, angelehnt an das englische „stencil“ für „Schablone“. Bei dieser Technik werden Schatten mithilfe des Stencil-Buffers aus der Levelbeleuchtung sozusagen herausgeschnitten. Der Stencil-Buffer dient dazu, bestimmte Bildschirmelemente vor dem Rendern auszusparen, HUD-Elemente etwa.

Stencil Shadows basieren auf der Projektion der 3D-Modelle und haben daher quasi scharfe „Schnittkanten“. Außerdem sind sie pixelgenau und werden in voller Auflösung berechnet, sie neigen daher nicht zum Flimmern. Sie benötigen außerdem nicht viel Grafikspeicher.

## Stencil-Shadow-Engines

Auch andere Spiele nutzen Stencil Shadows, darunter der Horror-Shooter FEAR. Dort versuchte man, mit einem Trick eine Art Schattenfilterung zu nutzen, obwohl dies mit der Technik im Grunde nicht möglich ist: Mehrere Stencil Shadows wurden übereinandergelegt, um so eine Illusion von „soften“ Schatten zu erzeugen. Dies ließ die Hardware-Anforderungen des sowieso schon sehr anspruchsvollen Spiels jedoch massiv in die Höhe schnellen und war damit nur wenig praktikabel. Neben id Software (Doom 3) und Monolith (FEAR), versuchte sich außerdem Crytek an Stencil Shadows und nutzte diese im ersten Far Cry. Die Blütezeit der Stencil-Shadows ist eng verbunden mit dem Siegeszug des Per-Pixel-Lightings. Engines wie die id Tech 4, Cryengine 1, Unreal Engine 3 oder Source Engine 2 kehrten sich von dem zuvor verbreiteten Vertex-Lighting ab und setzten auf die verfeinerte, pixelgenaue Beleuchtung. Mit der verbesserten Beleuchtung wurden auch neue Berechnungen für die dynamische Schattendarstellung nötig.

## Stencil-Shadows-Nachteile

Weiche Schatten und die Berücksichtigung von Alpha-Texturen sind mit Stencil-Shadows allerdings nicht möglich. Und es gibt weitere Probleme: So kann es zu Fehldarstellungen kommen, wenn sich die Kamera komplett im verschatteten Bereich befindet, die Schatten wirken häufig perspektivisch inkorrekt und es können Löcher bei der Projektion der 3D-Modellen entstehen. Zudem können Transparenztexturen nicht berücksichtigt werden. Alpha-Texturen sind heutzutage allgegenwärtig und kommen für Vegetation, feine Strukturen wie Zäune oder Gitter, aber auch Haar oder andere Details zum Einsatz, für die komplexe Polygonmodelle nicht zweckdienlich sind.

## Shadow Mapping

Die damals zum Einsatz kommenden Stencil Shadows unterscheiden sich recht stark von dem heute gängigen Shadow Mapping. Diese Technik nutzt Shadow Maps, also (Graustufen-)Texturen, um die Schatten darzustellen und kommt in praktisch allen aktuellen Titeln zum Ein-

satz. Diese Technik bietet Vorteile bei Transparenzen, sie benötigt nur einen einzelnen Renderpass (ab DX 10) und aufwendige Filtertechniken können genutzt werden.

In aktuellen Spielen können hunderte Alpha-Texturen zum Einsatz kommen. Dabei wären einige Schatteneffekte mit der Stencil-Technik aus Doom 3 so nicht möglich. Ein Grund: Beim Shadow Mapping werden auch die Transparenztexturen abgetastet, zudem können selbst Halbtransparenzen wie beispielsweise Partikelrauch berücksichtigt werden. Die Schatten können außerdem mit weichen Kanten versehen und aufwendig gefiltert werden, was einen insgesamt deutlich realistischeren Look erlaubt. Aus diesem Grund hat sich in aktuellen Spielen diese Technik durchgesetzt. Doch auch das Shadow Mapping hat handfeste Nachteile.

## Nachteile Shadow Maps

Die Darstellung von Shadow Maps ist sehr rechenaufwendig, denn grundsätzlich muss für jeden Pixel der zum Einsatz kommenden Schattentexturen einzeln berechnet werden, ob an dieser Stelle ein Schatten vorliegt und wie dunkel er ausfällt. Auch muss die Schattentextur im Speicher der Grafikkarte abgelegt werden. Je größer die Schattentextur, desto mehr muss die Hardware rechnen und desto mehr Speicher wird benötigt. Deshalb sind die Ausmaße dieser Texturen häufig stark begrenzt. In großen Spielewelten muss die Textur daher über einen weiten Raum gestreckt werden, was zu einer niedrigen Pixeldichte und damit Flimmern und Verzerrungen führt. Shadow Maps neigen daher zum Flimmern, was wiederum eine Filterung nötig macht, die ebenfalls viel Leistung kosten kann.

Shadow of the Tomb Raider (hier mit Shadow Banding mit RT „Hoch“) erlaubt einen Blick in die Zukunft: Hier werden in Echtzeit berechnete „Raytracing“-Schatten erzeugt; diese spiegeln die realen Lichtverhältnisse wieder und wirken sehr realistisch – kosten allerdings auch viel Leistung

Deshalb nutzen Spiele abgestufte (cascaded) Shadow Maps. Das heißt: In der unmittelbaren Umgebung kommt eine kleine, aber hochauflösende Schattentextur zum Einsatz. In mittlerer Distanz dann eine größere, aber mit insgesamt niedrigerer Pixeldichte. In der Ferne wird eine weitere Textur dann schließlich weit gestreckt und nur in niedriger Auflösung berechnet. Diese Abstufung lässt sich in praktisch allen Spielen beobachten.

## Tricksereien

Neben den abgestuften Schattentexturen kommen auch noch weitere Tricksereien und Vereinfachungen zum Einsatz. Ein großes Hindernis bei der Entwicklung ist die maximale Anzahl gleichzeitig nutzbarer Lichtquellen, die überhaupt einen Schatten werfen können.

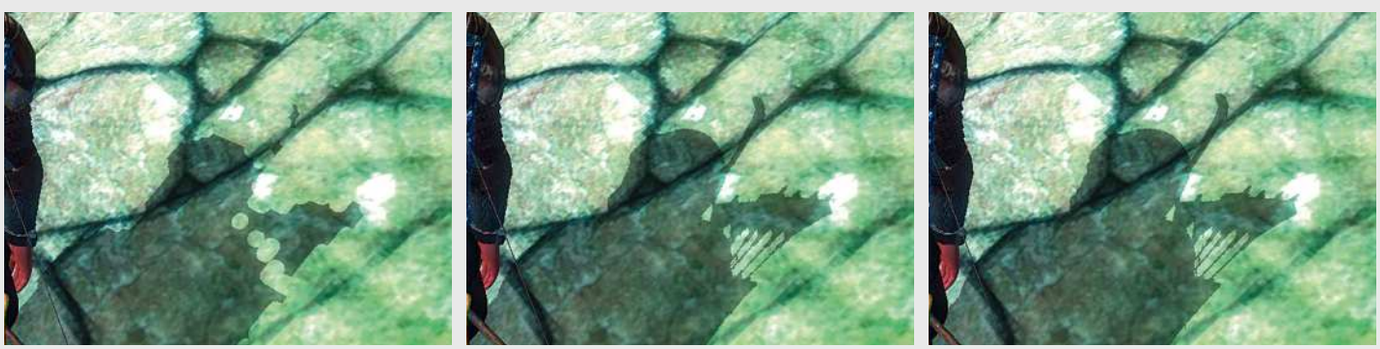
Für jede einzelne Lichtquelle ist mindestens ein Renderdurchlauf der Grafikkarte pro Frame nötig, zudem erhöht sich mit jeder zusätzlichen Lichtquelle die Anzahl Renderanweisungen (Draw Calls), welche der Prozessor an die Grafikkarte versenden muss. Dies kann dazu führen, dass Schatten nicht nur die Grafikkarte belasten, sondern die komplette Kommunikation zwischen den Hardware-Komponenten blockieren.

## Schatten und die Leistung

Schatten gehören zu den wichtigsten Grafikeffekten in modernen Spielen, sind aber auch heute noch unter den leistungsfressendsten Optionen im Grafikmenü. Neben der Schattenauflösung und Sichtweite sind vor allem „Extras“ wie eine hochqualitative Filterung sehr teuer. Es lohnt sich zumeist, bei diesen Einstellungen ein wenig Zurückhaltung zu üben.

# Wie Shadow-Maps die Qualität bestimmen

**Auflösung von Schattentexturen im direkten Vergleich: Während die schattenwerfende Geometrie – hier die Spielfigur mitsamt Ausrüstung wie dem Pfeilköcher – immer in derselben Auflösung vorliegt, variiert die Auflösung der Schattentextur von 128 über 1.024 bis 2.048 Pixel pro Kante von links nach rechts. Gut zu erkennen: der hohe Detailgewinn bei den Schatten zwischen 128 und 1.024 Pixeln.**



# Stencil-Shadows

Mit Doom 3 machte sich Engine-Guru John Carmack ein weiteres Mal von sich reden. Die „Carmack's Reverse" getaufte Schattentechnik nutzt den Stencil Buffer. Der Stencil Buffer dient normalerweise dazu, Teile des Bilds vor dem Rendern auszuschließen, beispielsweise unter einem HUD verborgene Elemente, die unnötig Rechenzeit benötigen würden – „Stencil" bedeutet im Deutschen „Schablone", Shadow Volumes nennt sich die zugrundeliegende Projektion auf Basis der Vertices der Polygonmodelle: Von jeder Schatten werfenden Lichtquelle (Doom 3 nutzt maximal zwölf zugleich) wird der Lichtkegel verfolgt, von sich in diesem Kegel befindliche Objekte wird eine auf dem Polygon-Gerüst basierende Silhouetten-Projektion kreiert und in den Stencil Buffer geschrieben. Die Vorteile der Stencil Shadows sind unter anderem, dass sie kaum (zusätzlichen) Videospeicher benötigen, saubere Kanten haben und in nativer Auflösung pixelgenau appliziert werden. Nachteile: Da Shadow Volumes auf Polygonstrukturen basieren, können Alpha-Texturen (z.B. für die Darstellung von Vegetation) nicht berücksichtigt werden, es sind mehrere Renderpasses (GPU-Berechnungen) pro Frame, Schatten und Lichtquelle nötig, die CPU-Last ist tendenziell hoch und gefilterte sowie halbtransparente Schatten funktionieren nicht – Stencil Shadows werden entweder angezeigt oder eben nicht. Außerdem fordern sie der GPU eine hohe Füllrate ab und es gibt Perspektivenprobleme.

Die aus Shadow Volumes kreierten Stencil Shadows in Doom 3 waren damals extrem beeindruckend. Im Gegensatz zu Shadow Maps sind Stencil Shadows immer scharf

FEAR nutzt ebenfalls Stencil Shadows. Sogar „Soft Shadows" sind möglich

Doch es ist nur ein Trick: Mehrere Schatten werden versetzt übereinander gelegt

# (Cascaded) Shadow Mapping

In aktuellen Titeln wird beinahe ausschließlich Shadow Mapping verwendet. Shadow Maps basieren auf einer Textur, die eine bestimmte Größe und Auflösung einnimmt und die Sie im Grunde mit der Kamera bzw. Figur durch die Spielewelten bewegen. Dazu wird der „Blickwinkel“ jeder Schatten-werfenden Lichtquelle einzeln berechnet, die Szene dann nochmals aus der Kameraperspektive berechnet, potenzielle Hindernisse erfasst und dann für jeden Pixel der Schattentextur entschieden, ob und wie stark dieser verdunkelt werden soll – je höher die Auflösung der Shadow Map, desto anspruchsvoller ist dieser Vorgang also. Da eine einzelne, gestreckte Map nicht ausreichen oder man eine sehr ineffiziente Pixelgrößen benötigen würde, kommen mehrere, abgestufte (cascaded) Texturen zum Einsatz: In der Nähe mit höchster Pixeldichte dargestellt, mit zunehmender Entfernung werden die Maps flächenmäßig größer, die Auflösung sinkt. Vorteile von Shadow Maps: Pro Lichtquelle ist nur ein einzelner Renderpass nötig (ab DX 10), Alpha-Texturen und andere Transparenzen können berücksichtigt und ergänzt (z.B. Schatten werfende Partikel oder Wolkenschatten) sowie aufwendige Filterverfahren genutzt werden. Nachteile: Shadow-Maps sind flimmeranfällig, können Darstellungsprobleme mit sich bringen (Artefakte, Streifen) und benötigen viel Grafikspeicher. Letzteres gilt insbesondere, wenn aufwendig gefiltert wird und die Sichtweite hoch ausfällt (viele einzelne Shadow Maps).

Die Schatten in Battlefield 1 sind prinzipiell hübsch, aber etwas niedrig aufgelöst. Theoretisch wären auch halbtransparente Schatten, etwa für den Partikelrauch, möglich

Gut erkennbare Schattenkaskaden: Mit zunehmender Kameraentfernung sinkt die Schattenauflösung. Der Charakterschatten wird in Fallout 4 gesondert gerendert

# Sonnen- und Punktschatten

Auch Shadow Maps sind kompliziert zu berechnen und benötigen neben viel Speicher außerdem viel Rechenzeit. Neben abgestufen Schatten wird deshalb mit weiteren Tricks gearbeitet: So gibt es in aktuellen Spielen nur eine sehr begrenzte Anzahl Lichtquellen, die tatsächlich Schatten werfen – im Normalfall liegt das Limit mit heutiger Hardware und DX11-API bei acht bis zehn gleichzeitig Schatten werfender Lichter. Ansonsten wird das Spiel zu anspruchsvoll. Zudem steigt mit jeder Lichtquelle der CPU- bzw. API-Overhead. In vielen Spielen gibt es deshalb mehrere Schattenoptionen, oft getrennt für „Sonnenschatten“ und „Punktschatten“. Diese unterscheiden sich technisch im Grunde nicht allzu sehr. Die „Sonne“ ist im Prinzip nur eine einzelne, wichtige Lichtquelle, während andere Lichter wie Straßenlaternen oder die Beleuchtung in Innenräumen einfach ein wenig anders gehandhabt werden. In The Division ist dieser Unterschied allerdings recht auffällig, denn die Sonnenschatten (und die des Mondes, gleiches Prinzip) werden in deutlich höherer Qualität berechnet als jene, die von „künstlichen“ Lichtquellen geworfen werden. Zusätzlich nutzt The Division noch eine weitere Art der Schattendarstellung, die offenbar Screenspace-basiert arbeitet (die Schatten verschwinden am Rande des Blickfelds der Kamera) und zur hochauflösenden Verschattung von feinen, bodennahen Objekten wie diesem Kabel dient. Hellblade nutzt einen ähnlichen Ansatz für die

Diese Sonnenschatten sind deutlich höher aufgelöst, als jene künstlicher Lichter. Nachts kann man beide Schattenarten gleichzeitig beobachten (z.B. unter Laternen)

Die Punktschatten künstlicher Lichtquellen sind deutlich niedriger aufgelöst, zudem funktioniert die optionale PCSS- sowie HFTS-Darstellung nicht (hier zugeschaltet)

# Mainboards

# So funktioniert ein Mainboard: Überblick

Die Zahl der Schnittstellen „onboard“ scheint stetig zuzunehmen. Doch es verschwinden auch immer wieder Anschlüsse (zuletzt Firewire und eSATA)

Das Mainboard (auch „Hauptplatine“ oder „Motherboard“ genannt) ist die Schaltzentrale in jedem PC. Was der PC kann, hängt zum großen Teil von den Funktionen dieser Platine ab. Wie die einzelnen Bauteile funktionieren und welche Techniken dabei zum Einsatz kommen, zeigt das folgende Kapitel.

Bevor wir aber jede einzelne Schnittstelle und Funktion einzeln betrachten, wollen wir einen Überblick über die Grundfunktionen und den Aufbau der „Bretter“ geben.

## Vielschichtige Bretter

Auch wenn hunderte Bauteile auf einem aktuellen Mainboard sitzen, ist die darunter liegende Platine („Printed Circuit Board“, kurz: PCB) mindestens genauso wichtig wie die Masse der Chips, die auf ihr sitzen. Über das PCB werden alle Bauteile mit elektrischer Energie versorgt, auch der Datentransfer findet darüber statt. Damit dies möglich ist, ist die Platine aus mehreren Schichten (engl. „Layer“) aufgebaut, die unterschiedliche Aufgaben übernehmen.

Je komplexer eine Hauptplatine aufgebaut ist, desto mehr Schichten hat sie. Während viele Mittelklasse-Hauptplatinen aus sechs Lagen aufgebaut sind, handelt es sich bei Oberklasse-Mainboards häufig um acht Schichten: Die Anbindung von sechs RAM-Steckplätzen für den Triple-Channel-Betrieb stellt beispielsweise höhere Ansprüche an das Design als eine Dual-Channel-Anbindung mit vier Slots.

Selbst wenn sich Datenleitungen so verlegen lassen, dass ein Board mit weniger Platinenlagen auskommt, werden für eine stabilere Stromversorgung oder eine bessere Abschirmung der Schichten untereinander zur Minimierung elektromagnetischer Interferenzen manchmal mehr Schichten verwendet.

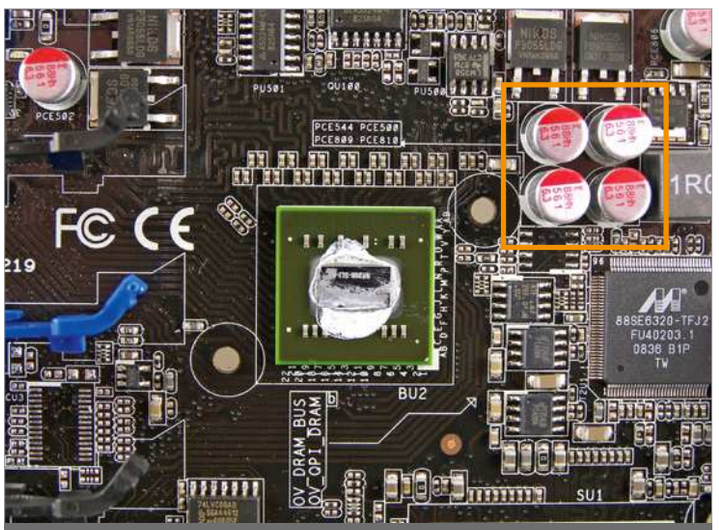
Je mehr Leitungen quer durch die Platine geführt werden müssen, desto komplizierter gestaltet sich Entwicklung und Fertigung, was unnötige Kosten verursacht. Außerdem sind die Signallaufzeiten bei kürzeren Strecken leichter zu kontrollieren; bei Frequenzen bis in den Gigahertz-Bereich ist jeder gesparte Zentimeter hilfreich für die Stabilität.

## CPU-Versorgung

Strom und Daten zu liefern, das ist also zunächst einmal eine wesentliche Aufgabe der Hauptplatine. Eine Komponente verdient aber noch besondere Beachtung: Die Stromversorgung des Hauptprozessors unterscheidet sich allein im Umfang deutlich von der anderer Komponenten.

In den letzten Jahren ist die Bedeutung der Spannungsversorgung gewachsen. Dies verwundert nicht, denn sie trägt maßgeblich zur Energieeffizienz, zur Langlebigkeit und zum Overclocking-Potenzial bei. Die Stromversorgung umfasst im Wesentlichen einen PWM-Controller für die einzelnen Phasen, MOSFETs samt Treiber, Kondensatoren und Spulen. Auch wenn allgemein nur von *der* CPU-Spannung die Rede ist, liegen innerhalb eines Prozessors verschiedene Spannungen für unterschiedliche Teilbereiche an.

Bei vielen aktuellen Prozessoren wird der Uncore- bzw. Northbridge-Teil der CPU unabhängig von




Auf der Epoxidharz-Platine eines Mainboards werden neben den elektronischen Bauteilen wie Kondensatoren (rechts) alle wichtigen Chips verlötet – in der Bildmitte zum Beispiel eine SLI-Chip

den Kernen versorgt. Aus diesem Grund werden die einzelnen Phasen getrennt genannt: Erfolgt die Stromversorgung etwa über „4+1 Phasen“, so sind von den insgesamt fünf Phasen vier für die Kernspannung und eine für den Uncore-/Northbridge-Bereich zuständig. Bei manchen CPUs kommt noch die Spannung für die integrierte Grafikeinheit und/oder weitere Komponenten hinzu.

## Phasen

Je mehr Phasen ein Mainboard hat, desto besser lässt sich die Last auf verschiedene MOSFET-Gruppen verteilen: Bei vier Phasen beträgt die effektive Arbeitszeit nur 25 Prozent, bei sechs liegt sie schon unter 20 Prozent. Damit das funktioniert, muss der Phasenwechsel zuverlässig funktionieren. Mehr Phasen sind zwar grundsätzlich, aber nicht zwangsläufig besser, da auch die Qualität der einzelnen Bauteile eine große Rolle spielt und bei zunehmender Komplexität Signal- und Timing-Probleme eher auftreten können. Auch die Energieeffizienz kann von einer überdimensionierten Stromversorgung in Mitleidenschaft gezogen werden. Bei manchen Mainboards wird die Phasenanzahl daher lastabhängig variiert.

## Der Chip(-satz)

Ein klassischer Mainboard-Chipsatz umfasst eine North- und eine Southbridge. Deren Namensgebung leitet sich von der üblichen Positionierung auf dem PCB ab – die Northbridge sitzt oben, nahe bei der CPU und dem RAM, die Southbridge befindet sich unten, bei den Steckplätzen für Erweiterungskarten und Laufwerksanschlüssen. Seit vielen Jahren verfügen die Mainboards aber nur noch über einen Baustein, der alle Funktionen übernimmt. Ein Großteil der ehemaligen Northbridge-Funktionalität, allen voran der Speichercontroller, wurde direkt in den Prozessorkern verlagert (siehe Kapitel dort).

Intel nennt den zentralen Chip auf dem Mainboard knapp „PCH“ („Platform Controller Hub“), bei AMD hieß dieser zuletzt „Fusion Controller Hub“, jetzt aber wieder „Chipset“. Dieser verbliebene Baustein wird meist entweder per PCI-Express (bei Intel DMI, Direct Media Interface genannt) oder Hyper Transport (AMD) an den Prozessor angebunden.

## Das BIOS

Beim „Basic Input Output System“ (BIOS) handelt es sich um die Firmware eines PCs, die auf einem EEPROM oder Flashspeicher abgespeichert ist und nach jedem Start des PCs ausgeführt wird. Noch bevor Windows geladen wird, sorgt es dafür, dass alle wichtigen Komponenten auf ihre Funktion überprüft und angesprochen werden können.

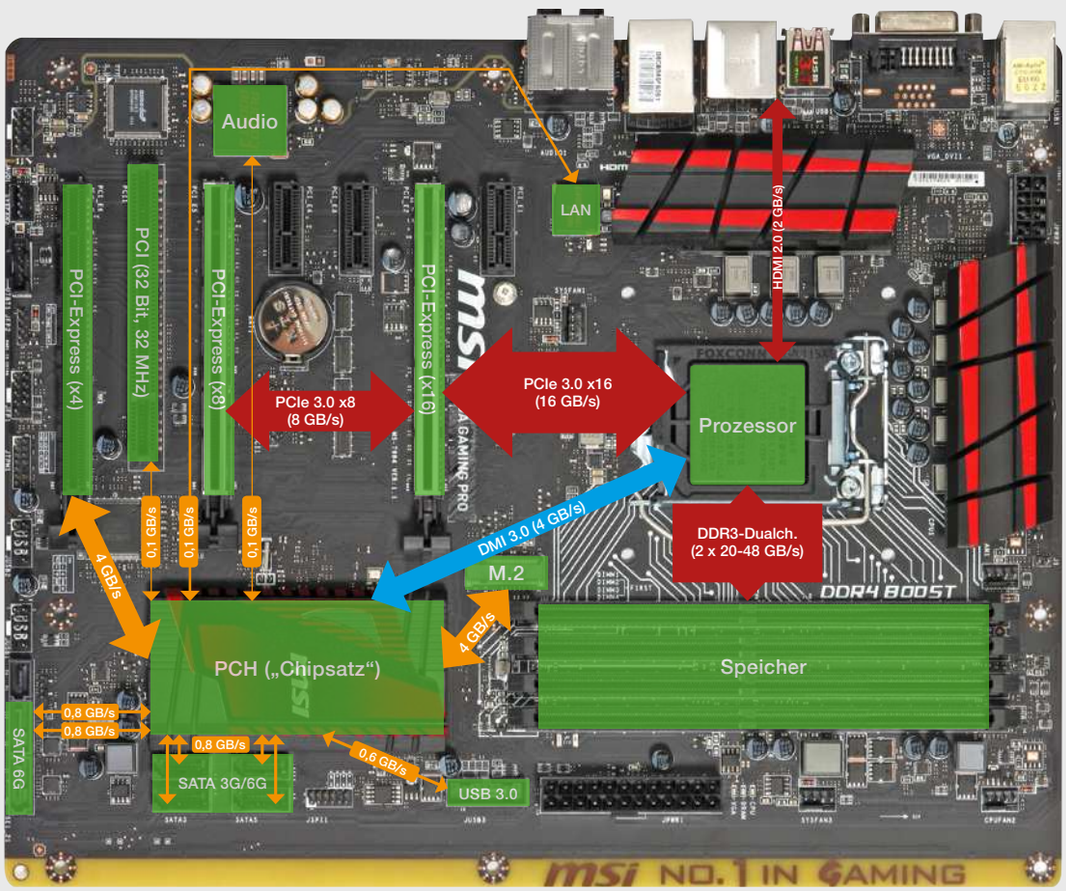
Der BIOS-Nachfolger (U)EFI erlaubt noch weit mehr Funktionen und sorgt für eine einfachere Bedienung. Mehr Informationen dazu sowie zu allen anderen Mainboard-Teilen finden Sie den folgenden Seiten.

# Datenströme im PC

Das Mainboard als zentrale Komponente verbindet nahezu alle Bauteile im PC miteinander. Die folgende (teils vereinfachte) Grafik soll die entstehenden Datenströme illustrieren und zeigen, wo potenzielle Engpässe auftreten können. Die Liniendicke entspricht dabei dem Datenvolumen.

Direkt mit dem Prozessor kommunizieren Speicher und PCI-Express-x16-Steckplatz – beide sind gleichzeitig die am Bandbreiten-hungrigsten. Außderm verfügen einige CPUs über direkte Verbindungen zur Grafik-Schnittstelle (rote Linie).

An den PCH angebundene Komponenten sind meist etwas weniger Ressourcen-hungrig (orangefarbene Linien). Dennoch kann die schiere Menge der Komponenten die Verbindung zur CPU (DMI) an ihre Grenzen bringen (blaue Linie).




Anm.: Board fiktiv; Angaben teils nicht kaufmännisch gerundet Illustration: eigene

# Die Mainboard-Bauteile im Detail

## PCI-Express-Slots

Von den PCI-E-Steckplätzen sind meist nur ein oder zwei auch elektrisch als x16-Variante ausgeführt. Die verbleibenden sind x1-, x2- oder x4-Steckplätze. Bei einigen Platinen gibt es außerdem noch einen PCI-Steckplatz für ältere Karten, diese Platine verzichtet allerdings darauf

## Batterie

Die Batterie sorgt dafür, dass bestimmte Daten wie die Uhrzeit und das Datum nicht bei jedem Start erneut eingegeben werden müssen

## Taktgeber

Ausgehend von einem 14,318-MHz-Oszillator werden alle Takte im PC von einem Taktgenerator-IC erzeugt. Dieser Baustein ist auch für flexible OC-Optionen verantwortlich

## M.2-SSD

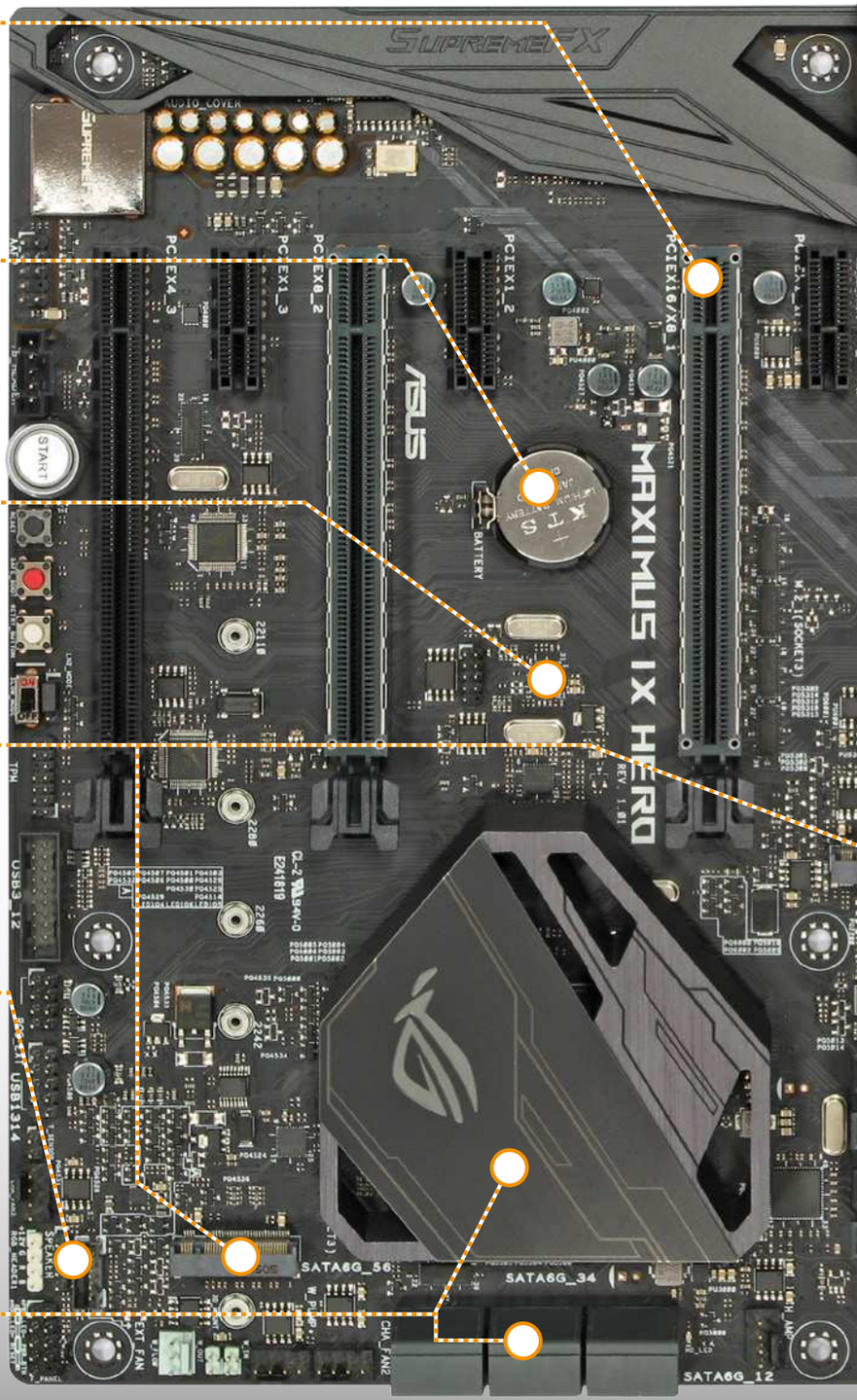
Da Flash-basierte SSDs sehr kompakt sind, können diese direkt auf dem Mainboard „angesteckt“ werden. Voraussetzung ist ein sogenannter M.2-Anschluss

## Anschlussfeld

Hier werden der An- und Aus- sowie der Resetknopf des Gehäuses und Zustands-LEDs, die zum Beispiel die Festplattenaktivität anzeigen, angeschlossen

## I/O-Hub („Chip“)

Nach der Lage auf dem Schaltplan wurde dieser Baustein früher „Southbridge“ genannt. Heute gibt es aber keine zugehörige „Northbridge“ mehr und aus dem „Chipsatz“ ist ein schlichter „Chip“ bzw. „PCH“ (Intel) geworden. Dieser stellt immerhin noch Anschlüsse für PCI-Express, USB und SATA bereit. Außerdem kontrolliert der Baustein den PCI-Bus, Audiofunktionen, Firewire, LAN sowie alte Schnittstellen über den LPC-Bus

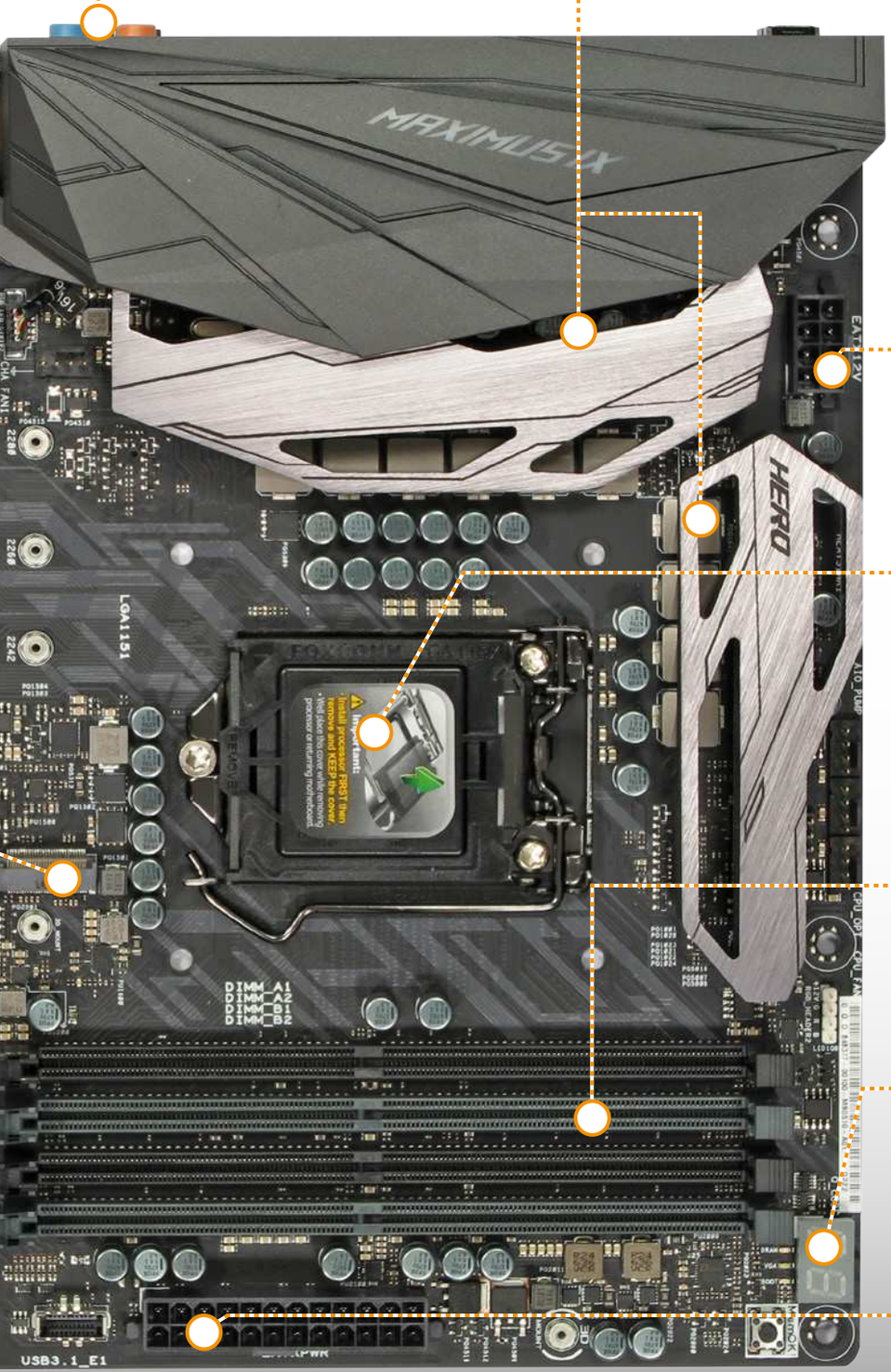


## Externe Anschlüsse

Neben Eingabegeräten (PS/2, USB) lassen sich hier unter anderem Plug&Play-Datenspeicher (USB/E-SATA), Netzwerk-Hardware (LAN) und Audio-Geräte anschließen

## CPU-Stromversorgung

Für die Stromversorgung der CPU inklusive der integrierten Komponenten wie System Agent, Grafikkern, Speichercontroller etc. kommen meist mehrere Polymerkondensatoren und Ferritkernspulen sowie 16 bis 24 MOSFETs zum Einsatz (hier unter einer Zierblende verborgen)




## Stromanschluss

Mit Energie wird das Mainboard über einen 24-poligen ATX-Stromanschluss und ein separates, 8-poliges ATX-12-Volt-Anschlussfeld versorgt

## CPU-Sockel

Der Sockel stellt die Verbindung zum Prozessor über Kontakte her; üblich sind mehr als 1.000 Stück bei aktuellen Modellen. Der Großteil davon wird für die Leitungen zum Speicher benötigt

## DDR3/4-RAM-Slots

Für den Dual-Channel-Betrieb muss ein Speicherriegel-Paar in gleichfarbige Slots eingesetzt werden. Maximal lassen sich meist 4 x 16 GiByte nutzen. Der Arbeitsspeicher ist direkt an die Northbridge bzw. die CPU angebunden

## Diagnose-LEDs

Einige Platinen verfügen über Diagnose-LEDs, welche den aktuellen Status-Code des Boards ausgeben. Dies kann bei der Fehlersuche helfen

## Stromanschluss

Mit Energie wird das Mainboard über einen 24-poligen ATX-Stromanschluss und ein separates, 8-poliges ATX-12-Volt-Anschlussfeld versorgt

# Mainboard: Externe Anschlüsse

## Sonderfunktionen

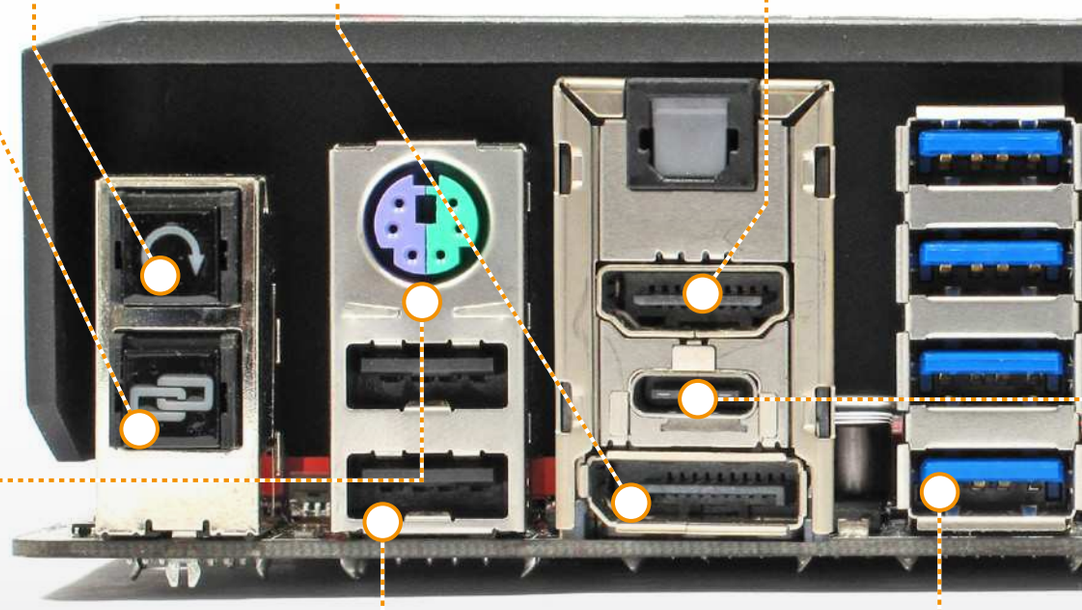
Vor allem auf teureren Mainboards finden sich manchmal Schalter für besondere Funktionen. So kann man beispielsweise ein Backup-BIOS einspielen, was vor allem für Übertakter sehr nützlich ist und bei fehlgeschlagenen BIOS-Updates hilft. Bei Asus darf man sich zudem über „ROG Connect“ freuen, das die ferngesteuerte Übertaktung des Rechners im laufenden Betrieb ermöglicht. Ebenfalls häufiger anzutreffen sind WLAN-, Bluetooth- oder Rest-Taster

## Displayport

Neuere Mainboards verfügen zusätzlich über einen Displayport, der unter anderem einen Kopierschutz unterstützt. Für die Displayport-Schnittstelle gibt es mehrere Spezifikationen, die aktuelle ist Version 1.4. Früher übliche D-Sub-Buchsen sind von der I/O-Blende des Mainboards praktisch verschwunden, DVI-Buchsen noch häufiger anzutreffen

## HDMI

Seit Prozessoren einen integrierten Grafikkern mitbringen, ist auf fast jeder Hauptplatine ein Monitoranschluss anzutreffen. Fast immer dabei: eine HDMI-Buchse, mit der sich die meisten Flachbildschirme der vergangenen Jahre betreiben lassen – aber nicht alle!



## PS/2

Viele Platinen haben mittlerweile gar keine PS/2-Ausgänge mehr, auch wenn die vollständige Verdrängung der Schnittstelle bis heute auf sich warten lässt. Gelegentlich wird ein kombinierter Anschluss verbaut, der die Maus-Funktionalität auf zwei normalerweise ungenutzte Pins im Tastatur-Anschluss legt. Zwei separate PS/2-Anschlüsse – getrennt für Maus und Tastatur – finden sich nur noch selten

## USB 2.0

Der heute wichtigste Anschlusstyp am I/O-Panel eines Mainboards sind externe USB-Ports. Vor rund zwei Jahrzehnten eingeführt, ist USB heute der bevorzugte Anschluss für fast jede Form externer Peripherie. USB-2.0-Anschlüsse bieten dabei eine Maximalgeschwindigkeit von 480 MBit/s (60 MByte/s) – effektiv bleibt unter anderem wegen der Codierung viel weniger übrig

## USB 3.0

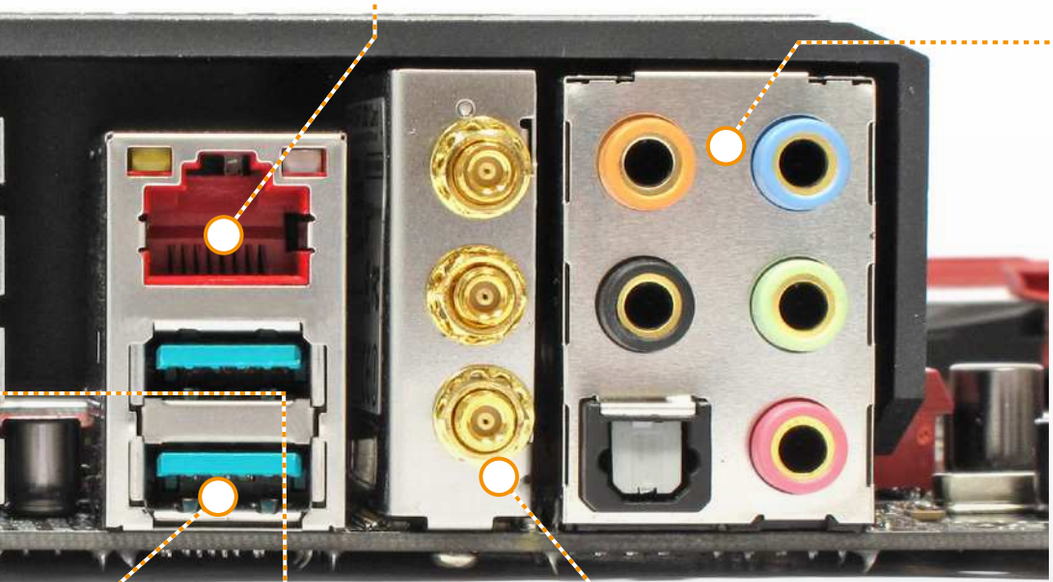
Die Nachfolge-Version 3.0 von USB ist nun schon einige Jahre auf dem Markt. Um entsprechende Anschlüsse von den älteren 2.0-Ports unterscheiden zu können, werden diese bei nahezu allen Herstellen farblich abgesetzt (oft in der Farbe Blau). USB 3.0 bietet eine maximale Geschwindigkeit von 5 GBit/s (625 MByte/s), von der effektiv rund 500 MByte/s bleiben, und dies natürlich auch nur dann, wenn das USB-Gerät so schnell Daten liefern kann

## USB 3.1

Der scheinbar kleine Versionssprung von 3.0 auf 3.1 sorgt immerhin für eine Verdoppelung der Datenrate (10 GBit/s, 1.250 MByte/s). Auf Mainboards sind solche Anschlüsse meist ebenfalls farblich ausgezeichnet (hier: türkis, manchmal auch rot). Leider gibt es keine einheitlichen Standards für die Farbe und zudem viel Verwirrung um neue Übertragungsmodi sowie Steckertypen. Im USB-Abschnitt weiter hinten im Kapitel bemühen wir uns um Aufklärung

## LAN

Als Gigabit-LAN-Controller günstiger wurden und irgendwann für wenige Cent zu haben waren, begannen die Mainboard-Hersteller, ihre Boards mit entsprechenden Anschlüssen vollzustopfen – nahezu jede Platine brachte zwei LAN-Anschlüsse mit. Mittlerweile ist etwas Vernunft eingekehrt und die meisten Platinen beschränken sich auf einen Anschluss. Der Platz wird heute eher für USB- und Displayanschlüsse verwendet



## Soundanschlüsse

Mit der Verbreitung von Onboard-Sound sind auch Audio-Aus- und Eingänge auf Mainboards gewandert. Typisch ist heute ein Funktionsblock mit fünf analogen Anschlüssen und einem optischen Ausgang, einige Mainboards nutzen aber auch sechs 3,5-mm-Klinke-Buchsen für getrennten Line- sowie Mic-In und platzieren den oder die digitalen Anschlüsse separat. Auf günstigen Mainboards sind dagegen oft nur drei Buchsen verbaut, dank flexibler Port-Konfiguration der meisten Onboard-Chips können daran wahlweise ein 5.1-System oder eine Kombination aus Stereo-Ausgabe- und Eingabe-Geräten angeschlossen werden. Die Farbcodes sind normiert und haben die folgende Bedeutung:

**Eingang für externe Quellen (AUX)**

**Ausgang für Frontlautsprecher oder (alternativ) Kopfhörer**

**Eingang für das Mikrofon**

**Ausgang für Center-Lautsprecher/Subwoofer (bei 2.1-, 4.1-, 5.1- oder 7.1-Systemen)**

**Ausgang für hintere Lautsprecher**

**Optischer Ausgang**

## USB 3; Typ C

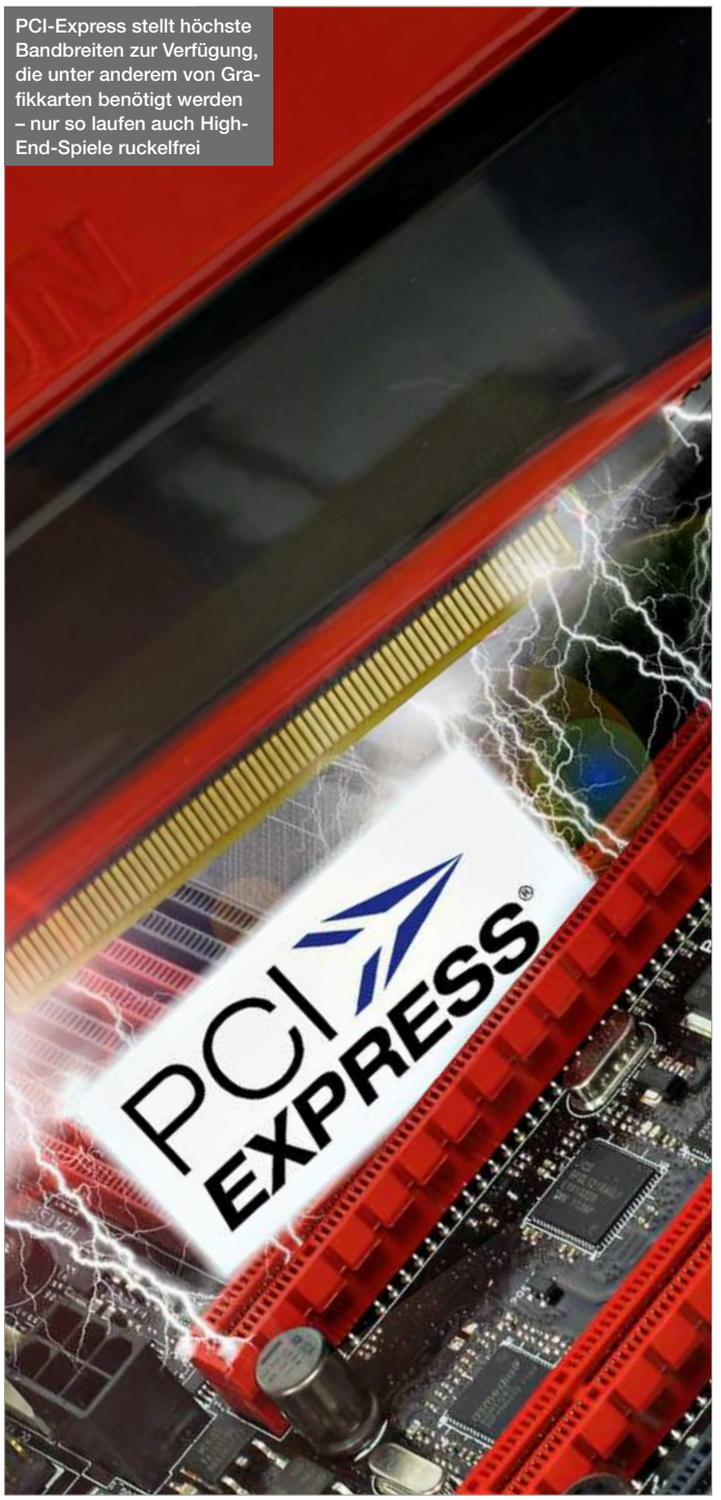
Weiter geht's im USB-„Wahnsinn": Ein neuer Steckertyp (der auch bei neueren Smartphones zu finden ist) soll Anschlussprobleme künftig verhindern. Es gibt zum Beispiel kein verkehrtes Einstecken mehr – der Stecker funktioniert immer, egal, wie herum man ihn anschließt. Was technisch drinsteckt, weiß man aber nicht – von USB 1.0 bis sogar „artfremden" Schnittstellen wie dem Displayport ist alles möglich

## WLAN

Auch Wireless-LAN braucht Kabel – nämlich vom Controller zur Antenne. Entsprechend ausgestattete Mainboards verfügen deswegen über entsprechende Antennen-(Kabel-)Anschlüsse am I/O-Panel

*Hypothetisches Mainboard mit nahezu allen verfügbaren Anschlüssen

# So funktioniert ein Mainboard: PCI-Express




PCI-Express stellt höchste Bandbreiten zur Verfügung, die unter anderem von Grafikkarten benötigt werden – nur so laufen auch High-End-Spiele ruckelfrei

Seit knapp 16 Jahren ist PCI-Express die wohl wichtigste Schnittstelle auf der Hauptplatine. PCI-Express gilt als Nachfolger zweier Techniken, nämlich der AGP-Schnittstelle für Grafikkarten und des PCI-Busses für alle anderen Erweiterungskarten.

Doch nicht nur Erweiterungskarten, sondern auch Zusatzcontroller, externe Peripherie, interne Laufwerke und andere Komponenten greifen immer häufiger auf die Schnittstelle zurück. Die Flexibilität von PCI-E wird jedoch intensiver genutzt als je zuvor, was in dieser Form eigentlich nie geplant war. Doch der Reihe nach.

## PCI und PCI-Express

Um zu verstehen, wie PCI-Express funktioniert, ist ein Blick in die Vergangenheit hilfreich: Der PCI-Bus hatte sich über eine ganze Dekade, etwa von 1994 bis 2004, bewährt und brachte unzählige Erweiterungskarten hervor.

Die ursprüngliche Spezifikation wurde von der PCI-SIG ausgearbeitet, einem 1992 gegründeten Konsortium, bestehend aus führenden Unternehmen im IT-Bereich (AMD, Compaq, Hewlett-Packard, IBM, Intel, Microsoft, Phoenix Technologies, Serverworks and Texas Instruments). Es verwaltet und kontrolliert alle Aktivitäten, Entwicklungen und Spezifizierungen rund um den PCI-Bus und dessen Nachfolger.

Ein riesiger Vorteil und eines der Hauptanliegen bei der Entwicklung von PCI-Express war es, dass sich das Software-Modell nicht vom klassischen PCI-Bus unterscheidet. Für das Betriebssystem sind demnach keine PCI-Express-Treiber nötig, anders als es etwa bei AGP der Fall war, wo ein spezieller GART-Treiber nötig war (ein Umstand, der oft für Stabilitätsprobleme verantwortlich war).

## Schnellere Geräte verlangen schnellere Schnittstellen

Doch es ging nicht nur um die Treiber: Zum Ende der Sockel-A-/478-Ära wurden die Limits von PCI überdeutlich. PCI taktete seine 32 Datenleitungen noch immer mit 33 MHz. Versuche, die Leistungsfähigkeit durch höhere Taktraten und einen breiteren Bus zu steigern (PCI 66 MHz, PCI 64 Bit, PCI-X – letzterer ist nicht mit PCI-Express zu verwechseln), waren nur im Servermarkt erfolgreich gewesen. Auf Endkunden-Mainboards stieg der Bedarf für schnelle Anbindungen mit der Verbreitung von Gigabit-LAN- und SATA-Zusatzcontrollern unterdessen weiter.

Bei theoretischen Datentransferrate von bis zu 150 MByte/s war von etwa 2002 an bereits ein einziges Endgerät in der Lage, den PCI-Bus zu überlasten. In einer Bus-Architektur müssen die knappen Ressourcen aber für alle Endgeräte reichen. Spätestens wenn Nachladevorgänge von der Festplatte oder Netzwerktransfers mit der Ausgabe des Spielesounds über eine Soundkarte in Konflikt gerieten, wurde das Problem auch für Endnutzer spürbar.

## Flickschusterei

Um den PCI-Bus und die Southbridge wenigstens vom Netzwerkdatenverkehr zu entlasten, entwickelte Intel für seine 2003er-Chipsatzgeneration am Ende eine Spezialschnittstelle (CSA), die Gigabit-LAN-Controller direkt mit der Northbridge

verband. Den gleichen Weg hatten Grafikkarten mit AGP bereits Jahre zuvor beschritten.

## PCI-Express

Ein Nachfolger für dieses Sammelsurium von zu langsamen oder zu spezialisierten Schnittstellen wurde somit dringend benötigt: „3GIO“ (3rd Generation Input/Output) sollte die Probleme lösen.

2003 war es dann soweit: Aus 3GIO war mittlerweile „PCI-Express“ geworden und die ersten Mainboards und Grafikkarten erschienen, wenn auch zunächst nur in homöopathischen Dosen. Der Name „PCI-Express“ ist etwas irreführend, denn mit dem alten PCI-Bus hat diese Schnittstelle nur noch das Ansteuerungsprotokoll gemeinsam.

PCI-Express ist kein Bus mehr, sondern eine bzw. mehrere Punkt-zu-Punkt-Verbindungen, die über einen Switch skalierbar sind, ähnlich wie bei einem LAN. Aufgrund des seriellen Aufbaus kann PCI-Express auf deutlich höhere Taktfrequenzen zurückgreifen.

## Serielle vs. parallele Datenübertragung

Für PCI müssen 32 Datenleitungen (64 in den Servervarianten) nebeneinander auf dem Mainboard und den Steckkarten verlegt werden. Trotz zahlreicher Richtungsänderungen muss die Länge dieser Leitungen dabei immer identisch sein, denn die Zuordnung der parallel übertragenen Bits wird bei PCI durch den Sende- und Empfangszeitpunkt definiert. Kommen die gleichzeitig auf allen 32 Leitungen gesendeten Bits nicht auch zum gleichen Zeitpunkt beim Empfänger an, ist die Information unbrauchbar.

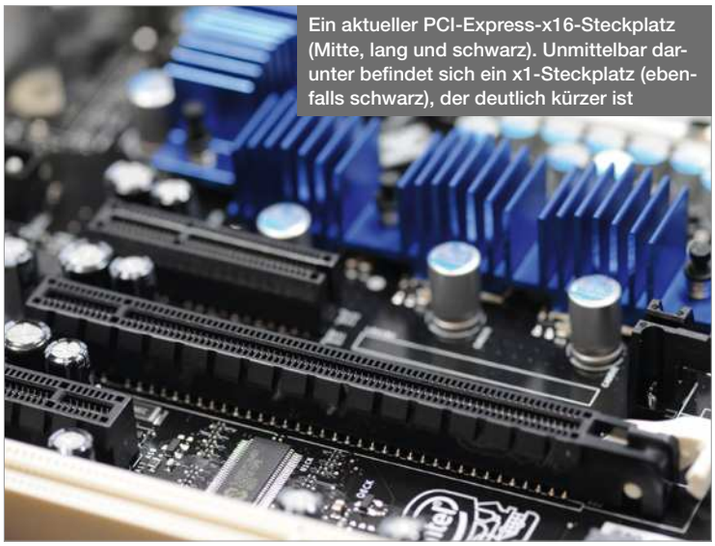
Leichte Unterschiede in der Datenlaufzeit sind allerdings kaum vermeidbar. Damit alle Bits innerhalb eines Taktzyklus eintreffen, ist die maximale Taktrate des Busses beschränkt.

Wie USB oder SATA umgeht PCI-Express dieses Problem durch eine serielle Übertragung. Alle Daten werden nacheinander auf der gleichen Leitung gesendet. Da die Laufzeiten auf der gleichen Ader zwangsläufig identisch sind, treffen die Daten somit automatisch in der richtigen Reihenfolge beim Empfänger ein.

Ein weiteres Problem für die Signalqualität war bei PCI das Bus-Konzept an sich. An offenen Endpunkten einer elektrischen Leitung und an unsauberen Verzweigungen sind Signalreflexionen möglich. Aufgrund der zahlreichen Abzweigungen zu ungenutzten Slots hat PCI von beidem reichlich, entsprechend stark war das Signalrauschen. Die ebenfalls zahlreich vorhandenen Empfänger dämpfen umgekehrt die Signalstärke insgesamt. PCI-Express entledigt sich beider Probleme schon konstruktionsbedingt: Die Leiterbahnen einer PCI-Express-Verbindung verlaufen ohne alternative Endpunkte von Host-Controller zum einzigen Empfänger.

## Takte und Leitungen

Natürlich muss der Takt dabei viel höher sein als bei einer seriellen Übertragung, um die „fehlenden“ Leitungen zu kompensieren. PCI-Express 1.0 arbeitet mit einer Grundfrequenz von 1,25 GHz, also einem Vielfachen des PCI-Taktes. Die Transferrate liegt somit theoretisch bei 2,5 GBit/s (1 Bit pro Hertz). Da die Übertragung jedoch im 8b10b-Code erfolgt, wo 2 Bit der Fehlerkorrektur dienen, bleiben unter dem Strich nur 2 GBit/s oder 250 MByte/s an Nutzbandbreite übrig – eine „Verschwendung“ von immerhin 20 Prozent, auf die wir später noch zurückkommen werden ...



Ein aktueller PCI-Express-x16-Steckplatz (Mitte, lang und schwarz). Unmittelbar darunter befindet sich ein x1-Steckplatz (ebenfalls schwarz), der deutlich kürzer ist

## Übertragungsgeschwindigkeiten im Vergleich

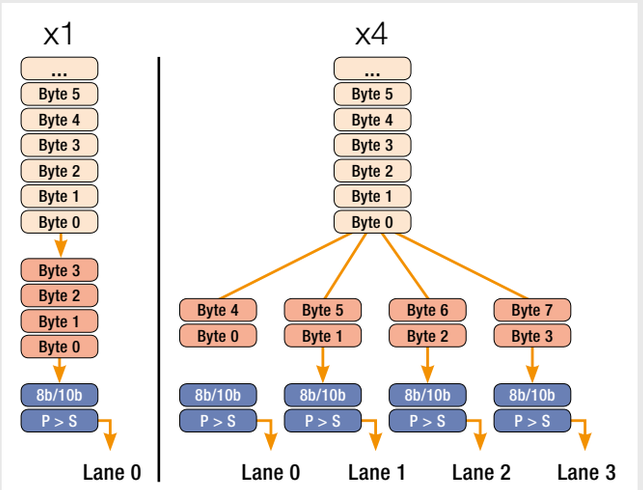
| | Jahr | Max. Takt | Busbreite** | Max. Datenübertragung |
|---|---|---|---|---|
| **ISA (XT-Bus)** | 1981 | 4,77 MHz | 8 Bit | 4,77 MByte/Sek. |
| **ISA (AT-Bus)** | 1984 | 8 MHz | 16 Bit | 16 MByte/Sek. |
| **EISA** | 1988 | 8,33 MHz | 32 Bit | 33 MByte/Sek. |
| **VLB (Vesa Local Bus)** | 1992 | 40 MHz | 32 Bit | 160 MByte/Sek. |
| **PCI 1.0** | 1991 | 33 MHz | 32 Bit | 133 MByte/Sek. |
| **PCI 2.x/3.0** | 1992 | 66 MHz | 64 Bit | 533 MByte/Sek. |
| **PCI-X 1.0** | 1998 | 133 MHz | 64 Bit | 1.066 MByte/Sek. |
| **PCI-X 2.0** | 2003 | 533 MHz | 64 Bit | 4.264 MByte/Sek. |
| **PCI Express 1.0 x1** | 2003 | 1.250 MHz | 8Bit/10 Bit | 250 MByte/Sek. |
| **PCI Express 1.0 x4** | 2003 | 1.250 MHz | 8Bit/10 Bit | 1.000 MByte/Sek. |
| **PCI Express 1.0 x8** | 2003 | 1.250 MHz | 8Bit/10 Bit | 2.000 MByte/Sek. |
| **PCI Express 1.0 x16** | 2003 | 1.250 MHz | 8Bit/10 Bit | 4.000 MByte/Sek. |
| **PCI Express 2.0 x1** | 2007 | 2.500 MHz | 8Bit/10 Bit | 500 MByte/Sek. |
| **PCI Express 2.0 x4** | 2007 | 2.500 MHz | 8Bit/10 Bit | 2.000 MByte/Sek. |
| **PCI Express 2.0 x8** | 2007 | 2.500 MHz | 8Bit/10 Bit | 4.000 MByte/Sek. |
| **PCI Express 2.0 x16** | 2007 | 2.500 MHz | 8Bit/10 Bit | 8.000 MByte/Sek. |
| **PCI Express 3.0 x1*** | 2011 | 4.000 MHz | 128Bit/130Bit | 985 MByte/Sek. |
| **PCI Express 3.0 x4*** | 2011 | 4.000 MHz | 128Bit/130Bit | 3.938 MByte/Sek. |
| **PCI Express 3.0 x8*** | 2011 | 4.000 MHz | 128Bit/130Bit | 7.877 MByte/Sek. |
| **PCI Express 3.0 x16*** | 2011 | 4.000 MHz | 128Bit/130Bit | 15.754 MByte/Sek. |
| **PCI Express 4.0 x16*** | 2017 | 8.000 MHz | 128Bit/130Bit | 31.508 MByte/Sek. |

* Wechsel der Datenkodierung von 8/10 Bit auf 128/130 Bit, dadurch höhere Netto-Ausbeute
** Bei PCI-Express: Fehlerkorrekturbits (Busbreite hier: jeweils 1 Bit für Hin- und Rückkanal)

Für Notebooks gibt es seit 2005 auch eine kleine Version von PCI-Express: Mini-PCI-Express findet sich aber nicht nur dort, sondern auch auf vielen Mini-ITX-Mainboards. Mit dem richtigen Adapter können auch vollwertige Erweiterungskarten angebunden werden

## Wie PCI-Express-Lanes Daten übertragen

**Die Daten werden bei PCIe seriell auf einer oder mehreren Lanes übertragen. Durch Zusammenschalten mehrerer Lanes wird die Transferrate ver-x-facht.**




Beim alten PCI-Bus wurden die Leitungen abwechselnd zum Senden und zum Empfangen von Daten genutzt („halbduplex“). Das ist in einem Bus hilfreich, wo verschiedene Geräte Sender und Empfänger sein können. Es erfordert aber auch ein aufwendiges System der Buszuteilung („Arbitrierung“), welches dafür sorgen muss, dass klar geregelt ist, welcher Teilnehmer an wen senden darf.

Da diese Arbitrierung weggefallen ist, sieht PCI-Express für jede Verbindung einen Hin- und einen Rückkanal vor, ein Leitungs*paar* also. Ein solches Leitungspaar heißt „Lane“ (vom englischen „Lane“, zu deutsch „Spur“ oder eben Leitung). Daten können simultan gesendet und empfangen werden. Wegen der gleichzeitig möglichen Übertragung auf dem Hin- und Rückkanal liest man oft auch von 500 GByte/s Gesamtbandbreite je (1.0-)Lane.

### Skalierung und Ausbaustufen

Mit den genannten Daten ist bereits die einfachste PCI-Express-Variante mit nur einer Lane bei erheblich geringerem Implementierungsaufwand deutlich schneller als der alte PCI-Bus. Für Geräte, die noch mehr Transferrate benötigen, etwa Grafikkarten oder RAID-Controller, können aber zusätzlich mehrere dieser Lanes zusammengefasst werden.

Daher gibt es für PCI-Express nicht nur eine Slot-Variante, sondern mit x1, x4, x8 und x16 gleich mehrere. Die höchste Ausbaustufe mit 16 „Lanes“ wird auch als „PEG“ („PCI-Express for Graphics“) bezeichnet. Eine kurze Bandbreitenrechnung: In einfacher Richtung befördert ein PCIe-x4-Gerät 1,0 GByte, eine PCIe-x8-Karte 2,0 GByte und ein PCIe-x16-Steckplatz 4,0 GByte Daten pro Sekunde über die Leitung – den Overhead schon abgezogen, und, wie erwähnt, nur in einer Richtung. Die gleiche Bandbreite steht jeweils auch in der anderen Richtung zur Verfügung!

### Designüberlegungen

Warum nicht gleich einen Slot mit der maximalen Lane-Anzahl? Für viele Anwendungen wäre diese Bandbreite überdimensioniert und außerdem war es eines der Ziele der PCI-SIG, eine möglichst kostengünstige Fertigung zu ermöglichen. Immer sind zumindest theoretisch alle Ausbaustufen miteinander kombinierbar, eine x1-Karte funktioniert immer auch in einem x16-Steckplatz – und es gibt noch viel mehr Möglichkeiten, dazu mehr auf der übernächsten Seite.

Noch ein Detail am Rande: Trotz der gleichzeitigen Nutzung mehrerer Datenleitungen handelt es sich bei PCI-Express nicht um eine parallele Schnittstelle. Jede Lane für sich arbeitet unabhängig und verschickt einzelne, in sich geschlossene Datenpakete. Zusammengefügt werden die Daten erst im Empfänger anhand der Sequenznummern der Pakete. Hierbei ist es unerheblich, in welcher Reihenfolge die Pakete eintreffen. Sogar ein erneuter Versand bei Übertragungsstörungen auf einer Lane ist möglich, ohne dass CPU oder GPU in ihrer Arbeit gestört werden.

### Weitere Ausbaustufen

Natürlich blieb es nicht bei der Ur-Version von PCI-Express. 2007 folgte die Revision 2.0 des Standards. Hier wurde die Grundfrequenz von 1,25 GHz auf 2,5 GHz verdoppelt, was auch zu einer Verdoppelung der möglichen Transferrate führte. PCI-Express-2.0-x16-Geräte (zum Beispiel Grafikkarten) können damit bis zu 8 GByte/s senden und empfangen, was einer Gesamtbandbreite (hin und zurück) von 16 GByte/s entspricht.

Dennoch ist PCIe 2.0 voll abwärtskompatibel zum PCIe-1.x-Standard. PCIe-2.0-Geräte können problemlos in PCIe-1.x-Slots eingesetzt werden, PCIe-1.x-Geräte in PCIe-2.0-Slots.

## PCI-Express 3.0

Seit 2011 ist PCI-Express 3.0 verfügbar bei Hauptplatinen und High-End-Grafikkarten. Im Vordergrund stand wieder die Verdoppelung der Datenübertragungsrate. Allerdings stellte eine weitere Steigerung um den Faktor 2 die PCI-SIG vor offenbar unlösbare Probleme: 10 Gigahertz über eine Steckverbindung zu übertragen, war unter den Vorgaben der kostengünstigen Bauweise und der Abwärtskompatibilität zunächst nicht in die Praxis umzusetzen. Für PCI Express 3.0 konnte man mit den herkömmlichen Methoden nur eine Taktfrequenz von 4 GHz erreichen – 60 Prozent mehr als im 2.x-Standard definiert.

Um die Transferrate noch weiter zu erhöhen, beschäftigte sich das Konsortium mit dem recht verschwenderischen Overhead, den 20 Prozent, von denen wir eingangs schon gesprochen haben. Anstelle des vorigen 8b10b-Schemas kommt nun ein Verfahren zum Einsatz, bei dem 130 Bit große Datenpakete übertragen werden, die jeweils 128 Bit Nutzdaten enthalten. Der Overhead sinkt auf rund 1,6 Prozent, die nutzbare Bandbreite steigt auf knapp 8 GByte/s in einer Richtung. Die Verdoppelung ist somit fast perfekt.

## PCI-Express 4.0

Nach einigen Verzögerungen steht nun auch PCI-E 4.0 vor der Tür – 2019 sollen erste Produkte erscheinen. Die Übertragungsgeschwindigkeit wird erneut gesteigert, dieses Mal auf knapp 16 GByte/s in einfacher Richtung. Damit die Signale bei der Taktfrequenz von nunmehr 8 GHz auch noch lesbar bleiben, wurden einige Parameter angepasst. So darf die Leitungslänge jetzt nur noch wenige Zentimeter betragen, während es vorher mehr als ein halber Meter war. Außerdem erfordert PCIe 4 bessere Steckverbindungen, was die Implementierung aber auch geringfügig teurer macht. Immerhin kommt die jüngste Generation der Schnittstelle nach wie vor ohne optische Datenübertragung aus.

## Stromtransport

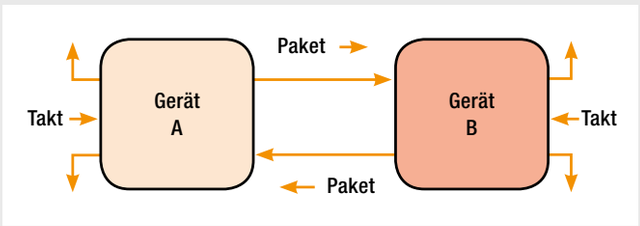
Ein Problem, das sich im Laufe der Jahre bei PCI und AGP entwickelt hatte, war die Stromversorgung. Als die Spezifikationen festgelegt wurden, hatte niemand daran gedacht, dass eine Grafikkarte 100 Watt oder mehr an elektrischer Leistung aufnehmen würde. Entsprechend wackelig konnte ein solches System in Sachen Stabilität sein, denn der PCI-Slot war lediglich für 25 Watt spezifiziert. Bei einem PEG-(PCIe-x16)-1.0-Slot sind es immerhin 75 Watt, was aber immer noch zu wenig ist.

Bei High-End-Grafikkarten sieht die Spezifikation daher unterschiedliche Zusatzstecker zur Stromversorgung vor, sogenannte PCI-Express (Graphics) Power Supply Connector (auch PEG-Connector), die +12V liefern. Die erste Version der Zusatzstecker hat 6 Pins und kann bis zu 75 Watt zusätzlich liefern, wodurch die dem Gerät maximal bereitgestellte Leistung auf 150 Watt steigt. Bei Nutzung zweier solcher Stecker sind sogar bis zu 225 Watt möglich.

In der Spezifikation von PCI-Express 2.0 wurde ein neuer Zusatzstecker mit 8 Pins definiert, der maximal 150 Watt führen kann. Für noch höhere Leistungen kann ein zusätzlicher Stecker mit 6 Pins genutzt werden, der jedoch nur weitere 75 Watt führt, wodurch die maximale Aufnahmeleistung einer PCI-Express-Karte auf insgesamt 300 Watt begrenzt ist.

# Wie eine PCI-Express-Lane aufgebaut ist

**PCI-Express kann auf separaten Leitungen in beide Richtungen gleichzeitig senden. Die Kombination der dafür nötigen Leitungspaare nennt man Lane.**




# Wie PCI-Express Übertragungsfehler verhindert

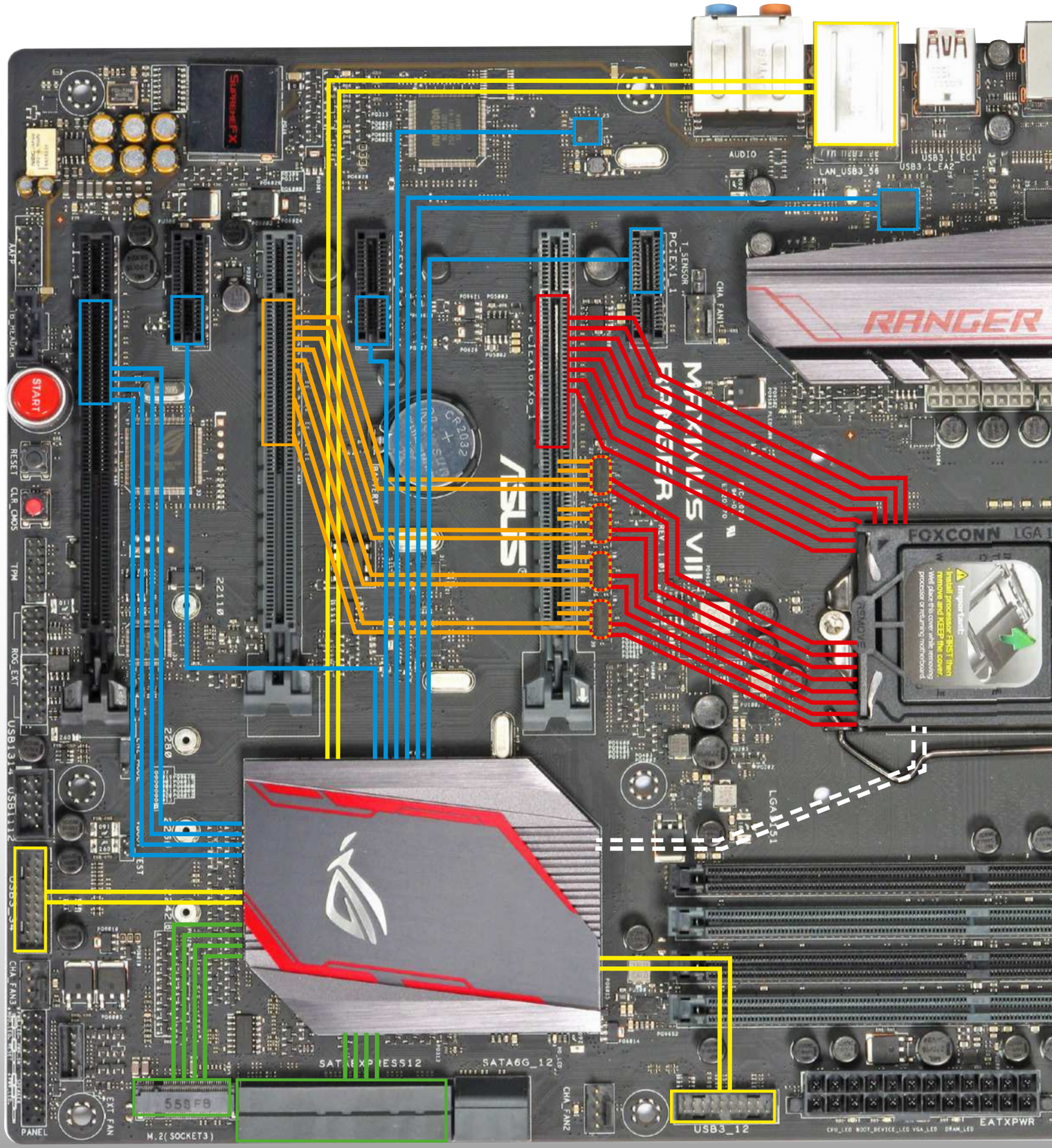
**Bussysteme wie PCI kämpfen mit verschiedenen Problemen: Schwierig ist es nicht nur, das gleichzeitige Ankommen der Daten sicherzustellen. Probleme bereitet auch die Signalqualität an sich. Die 32 Leitungen (bei PCI, siehe Abbildung rechts) beeinflussen sich gegenseitig, zudem kann es an jedem Zwischen- und Endpunkt der Bus-Topologie zu Signalreflexionenen kommen, welche die Signale weiter „verwaschen".**

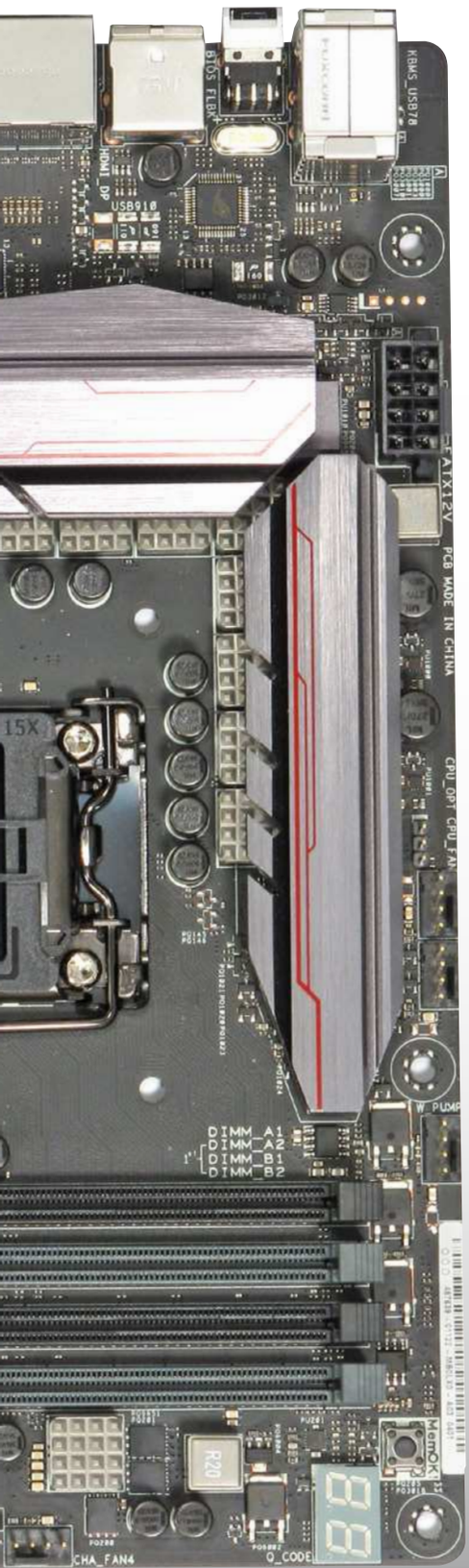
**Bei PCI-Express erfolgt die Übertragung eines Bytes dagegen mithilfe von zwei Leitungen: Welcher Wert übertragen wird, bestimmt nun nicht die Spannung auf einer Leitung wie bei PCI, sondern die Spannungs-*differenz* von zwei Adern. Der Vorteil: Verfälscht eine Störquelle das Signal, so wirkt diese immer auf beide Adern – die Differenz bleibt also gleich, auch wenn sich die absolute Spannung ändert.**

**Für eine Lane sind physisch also vier Leitungen notwendig: Je eine Hin- und Rückleitung, von denen jede wiederum zwei Adern enthält. Diese Tatsache führte gerade in der Anfangszeit häufig zu Verwirrung.**

**Für alle, die es ganz genau wissen wollen, noch eine Anmerkung zum Schluss: Streng genommen arbeitet auch PCI mit zwei Leitungen: einer spannungsführenden Leitung sowie der Masseleitung. Störquellen wirken hier aber nur auf die spannungsführende Ader, daher die schlechtere Signalqualität.**

# Überblick: PCI-Express auf dem Mainboard






## PCH/USB

Nicht nur der Prozessor verfügt über PCI-Express-Lanes; in aktuellen Architekturen stellen die „Hubs“, also das, was von North- und Southbridge übrig ist, weitere Leitungen zur Vernetzung bereit. Hier gelb markiert sind alle Verbindungen von und zu den USB-Anschlüssen (intern wie extern). Typischerweise wird ein USB-Controller bis 3.0 mit einer x1-Leitung angebunden, USB-3.1-Controller dagegen mit zwei Lanes.

## CPU/Grafik

Damit aber auch zwei Grafikkarten gleichzeitig genutzt werden können, integrieren viele Mainboards sogenannte PCI-Express-„Splitter“. Die Verbindung wird dann einmalig beim Einschalten geroutet, je nachdem welche Geräte gerade angeschlossen sind. Solche Splitter können nicht nur bei Grafikkarten zum EInsatz kommen, sondern grundsätzlich bei allen Lanes, die eine wechselseitige Nutzung erlauben.

## PCH/SATA & Co.

Für SATA-Geräte bis 6G hat der Chipsatz eigene Leitungen und ein eigenes Protokoll. Die Nachfolgetechniken SATA Express bzw. M.2 verwenden nun aber ebenfalls PCI Express als Transfermedium. Entsprechende Geräte werden meist mit vier oder zwei PCI-Express-3.0-Lanes angebunden.

## CPU/Grafik

Die meisten aktuellen Prozessoren verfügen über 16 PCI-Express-Leitungen zur direkten Anbindung der Grafikkarte. Im Normalfall steckt eine einzelne Grafikkarte im obersten Slot – diese erhält dann auch alle 16 Lanes.

## PCH/PCIe

Bei allen bandbreitenhungrigen Komponenten muss ein Mainboard auch noch Platz für diverse Einsteckkarten bieten – eine zweite Grafikkarte etwa oder Thunderbolt- und Soundkarten. Last but not least wollen Onboard-Bausteine wie Gigabit-LAN-Chips mindestens mit PCI-Express-3.0-x1 an den PCH angebunden werden. High-Speed-Chips wie Alpine Ridge benötigen sogar vier dieser Lanes.

## DMI

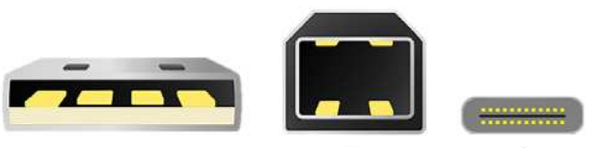
Der Prozessor ist in aktuellen PCs mit vielen wichtigen Komponenten direkt verbunden (Speicher, Grafikkarte, etc.) – aber nicht mit allen! Um die Daten des restlichen Systems zu übertragen, ist ein „I/O-Hub“ (auch „PCH“ oder „FCH“) mit eigenen PCI-Express-Lanes an den Prozessor gekoppelt. Da bei einigen Plattformen aber nur vier PCIe-x3-Lanes verwendet werden, ist dies ein potenzieller Flaschenhals – schließlich stellt der PCH seinerseits 20 Lanes mit je x1-Bandbreite bereit.

# So funktioniert ein Mainboard: USB

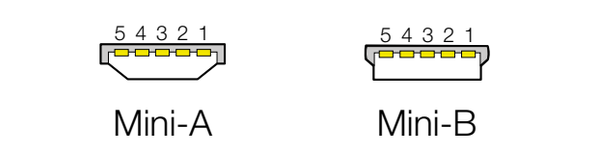
Nicht immer sind USB-Stecker gleich als solche zu erkennen: Rechts im Bild ein älterer USB-2.0-Mini-Anschluss, der kaum noch verwendet wird. Links der Micro-3.0-Standard, der die zusätzlichen Kontakte von USB 3 herausführt

## USB-Anschlusstypen: Typ A, B und C

**USB-Anschlüsse liegen mittlerweile in drei Varianten vor: Der Typ A (links) dürfte den meisten vertraut sein – er ist Quasi-Standard auf allen Mainboards. Etwas seltener kommt der Typ B (Mitte) zum Einsatz, in der Regel immer auf Geräteseite. Der Typ B hat bei USB-3.0-Geräten zusätzlich einen Aufsatz, damit er nicht in ältere Geräte gesteckt werden kann. Mit USB 3.1 kam ein weiterer Steckertyp hinzu: Typ C (rechts) ist kleiner und passt immer – egal, wie herum man ihn aufsteckt. Dies sind die „Fullsize"-Steckertypen.**




**Daneben gibt es noch jeweils „Mini"- und „Micro"-Varianten von den Steckern A und B. Der Micro-B-Stecker dürfte dabei den meisten bekannt sein: Er lädt dank EU-Regelung einheitlich (fast) alle unserer Smartphones.**




Über USB muss man nicht viel Worte verlieren: anschließen, abwarten und los – so war es in der Vergangenheit zumindest üblich. Doch mit der dritten Generation der Schnittstelle hat sich das geändert.

Mit USB 3.1 begann die Verwirrung und mit USB 3.2 wurde es noch schlimmer. Hauptgrund: Die Bezeichnung verrät nichts mehr über die Datenrate. Zudem wurden und werden in der Übergangsphase Standards zum Teil umbenannt. Und als wäre das nicht genug, rückt der Thunderbolt-Standard immer näher an USB heran ...

### USB-Generationen

Dabei war lange Zeit alles ganz einfach. Die USB-Schnittstelle ist in den vergangenen Jahren stetig weiterentwickelt worden. Der aus heutiger Sicht kaum noch vorstellbaren ersten USB-Variante mit Low- und Full-Speed-Geschwindigkeiten von 1,5 beziehungsweise 12 MBit pro Sekunde folgte schnell die zweite Generation, die immerhin schon 480 MBit/s übertragen konnte.

Die stark erhöhte Geschwindigkeit war neben der relativ leichten und preiswerten Integration einer der Hauptgründe für den USB-Erfolg. Als immer größere Datenmengen von Festplatten, USB-Sticks und Digitalkameras übertragen werden mussten, stieß aber auch die zweite Generation an ihre Grenzen. USB 3.0 verachtfachte 2008 die Datenrate auf 5.000 Mbit (5 Gbit) pro Sekunde und schaffte zumindest vorübergehend Entlastung.

USB 3.1 verdoppelte 2013 die Übertragungsgeschwindigkeit von 5 auf 10 GBit/s, also auf 1.250 MByte pro Sekunde. Die doppelte Geschwindigkeit wurde neben einer verbesserten Datenkomprimierung auch durch überarbeitete Kabel und Stecker erreicht. Und nun kam aufgrund ähnlicher Namen und Daten erstmals Verwirrung auf.

### Stecker und Schnittstelle – der Unterschied

Wer den USB-Durchblick behalten will, sollte zunächst zwischen der reinen Steckerform und der technischen Schnittstelle unterscheiden.

Die Steckerform, das ist beispielsweise der klotzige, rechteckige Typ A, den jeder kennt – praktisch alle USB-Stick haben einen solchen. Der Stecker kann aber auch der neue, verdrehsichere USB Typ C sein (siehe Kasten links). Mit der technischen Schnittstelle dagegen bezeichnet man eine USB-Version, etwa USB 2.0, und damit immer auch die zugrundeliegende Spezifikation. Umgekehrt ist eine Typ-C-Buchse kein Garant für die SuperSpeed-Plus-Geschwindigkeit von 10 GBit pro Sekunde. Der Port kann genauso gut lediglich USB 2.0 (oft bei Smartphones der Fall) oder USB 3.0 unterstützen – eine Übersicht der möglichen Kombinationen finden Sie in der Tabelle auf der nächsten Doppelseite.

In der Praxis heißt das auch: Vertraute Regeln, nach denen man beispielsweise bei einer blauen Buchse von USB 3.0 ausgehen darf, während die schwarzen in der Regel nur USB 2.0 unterstützen, sind passé und so nicht mehr gültig.

### Superspeed

Spätestens hier ist es sinnvoll, die Geschwindigkeit nicht mehr nur an der Spezifikation festzumachen; denn Bezeichnungen wie „USB

3.1“ wurden später aufgeweicht und durch Zusätze undurchsichtig gemacht. Als Merkmal für die Geschwindigkeit sollte besser der „Name“ dienen: Hi-Speed für 480 Mbit (USB 2.0), Super Speed für 5.000 Mbit (USB 3.0), Super Speed Plus für knapp 10.000 Mbit/s (USB 3.1) und Super Speed USB 20 Gbps für die jüngste Ausbaustufe mit 20 Gbit (USB 3.2).

## Universeller Stecker

Wie erwähnt, können Stecker und Spezifikation gemischt werden – zwar nicht beliebig, aber immer noch so, dass es für gehörige Verwirrung reicht. USB-3.1-Hardware kann beispielsweise mit einem älteren Typ-A-Stecker, aber auch mit dem neuen Typ-C-Stecker versehen werden.

Doch damit nicht genug. Ein USB-Typ-C-Stecker kann mit einer Reihe verschiedener Protokolle kommunizieren – mit einigen auch gleichzeitig. So kann dieselbe Verbindung nicht nur Daten in unterschiedlichen Geschwindigkeiten übertragen, sondern gleichzeitig bis zu 100 Watt an elektrischer Leistung liefern, wenn die Buchse die Spezifikation „USB Power Delivery“ mitbringt. Ebenso gut kann Displayport 1.3 mit bis zu 5.120 × 2.880 Pixel bei 60 Hertz und MHL unterstützt werden und die Typ-C-Buchse damit als Anschluss für externe Monitore herhalten. Dies wird als „Alternate Mode“ bezeichnet; die Pins sind also fähig, mehrere Zwecke zu erfüllen.

## Nutzung und Verbreitung

So ist es seit vielen Jahren zwar technisch möglich, ein Notebook etwa per USB-C an einen Monitor anzuschließen, während das Notebook über ein und dasselbe Kabel vom Monitor den Strom bezieht, ein Bildsignal an ihn sendet und den USB-Hub des Monitors betreibt. Tatsächlich aber nutzen nur wenige Notebooks (hauptsächlich Macbooks) USB-C auch als Ladebuchse und nur wenige Monitore haben eine entsprechende Buchse. USB-C als Dockingstation-Verbindung ist in wenigen Business-Notebooks zu finden, in der Consumer-Sparte wird eher alles noch konservativ separat angeschlossen.

## Thunderbolt

Die Fähigkeit des USB-Typ-C-Steckers, mehrere Funktionen zu übernehmen, kommt nicht von ungefähr. Diese stammt von der Thunderbolt-Technik ab.

Thunderbolt? Ja, denn diese Schnittstelle wurde gewissermaßen mit USB 3 verschmolzen. Um zu verstehen, wie das kam, müssen wir ein wenig zurückblicken. Thunderbolt geht auf eine Initiative von Intel zurück, das ab Herbst 2009 mit dieser Technik viele bis dato vorhandene Schnittstellen ersetzen wollte. Das Besondere damals: Neben einer sehr hohen Übertragungsgeschwindigkeit und einer integrierten Stromversorgung lassen sich auch Video- und Audiosignale für Displays übertragen. Die Signale werden dabei gewissermaßen verschränkt, so dass eine annähernd gleichzeitige Übertragung erfolgt (siehe Illustration rechts).

In der aktuellen Version 3 liefert Thunderbolt eine enorme Datenrate von 40 Gbit/s (5.000 MByte/s.). Die hohe Geschwindigkeit lässt sich beispielsweise dazu nutzen, um mehrere Bildschirme zu betreiben, was vor allem bei Docking Stations für (Business-)Notebooks sinnvoll ist.

Der eigentliche „Clou“ ist aber, dass der Standard seit Version 3 den Typ-C-Stecker von USB nutzt und auch die geforderten 100 Watt elektri-

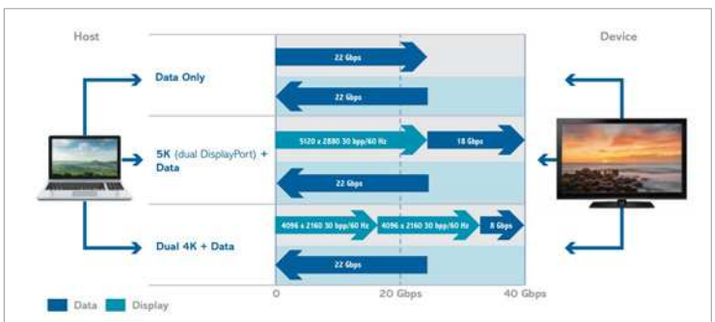
Bild: One Plus

Auch bei Handys muss man sich künftig leider wieder Gedanken um das Ladekabel machen. Der jahrelang omnipräsente Mini-B-Stecker wurde und wird gerade durch den Typ-C-Stecker (Bild) ersetzt

# Überblick: USB-Standards

| Name | Spezifikation | Brutto-Datenrate |
|---|---|---|
| **Low Speed** | USB 1.0 | 1,5 Mbit/s (187,5 kByte/s) |
| **Full Speed** | USB 1.0 | 12 Mbit/s (1,5 MByte/s) |
| **Hi-Speed** | USB 2.0 | 480 Mbit/s (60 MByte/s) |
| **Super Speed** | USB 3.0 / USB 3.1 Gen1 | 5.000 Mbit/s (600 MByte/s) |
| **Super Speed + (Plus)** | USB 3.1 Gen2 | 9.697 Mbit/s (1.212 MByte/s) |
| **Super Speed USB 20Gbps*** | USB 3.1 Gen2x2 | 19.394 Mbit/s (2.424 MByte/s) |

*manchmal auch „Super Speed ++“




Thunderbolt kann verschiedene Schnittstellen mit Daten versorgen und zusätzlich auch noch Bild- und Tonsignale übertragen.

## Wie Thunderbolt Daten überträgt

**Bei Thunderbolt können verschiedene Dateninhalte gemischt werden. So können nicht nur externe Geräte wie Massenspeicher angeschlossen werden, sondern auch Anzeigegeräte via Display Port.**

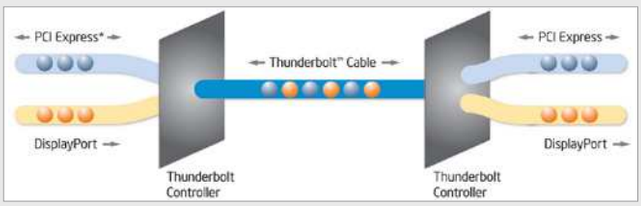



Bild: Intel

sche Leistung transportieren kann! Ein Thunderbolt-Controller kann also immer auch USB-Geräte ansprechen. Umgekehrt kann aber ein einfacher USB-C-Anschluss auf dem Mainboard nicht automatisch auch für Thunderbolt-Geräte genutzt werden. Es kommt immer auf den verbauten Controller an.

## Thunderbolt vs. USB

Thunderbolt 3 ist demnach zwar ähnlich, aber nicht identisch mit USB 3/Typ C. Die maximale Bandbreite von 40 Gbit/s ist außerdem (noch) Thunderbolt-exklusiv, USB muss sich mit 10 bzw. 20 Gbit/s begnügen. Zudem verlangt Intel Lizenzgebühren für Thunderbolt, weshalb die nötige Infrastruktur teurer ist als bei USB.

## Thunderbolt-Blitz

Wir halten fest: Am Typ-C-Stecker lässt sich also nicht mehr erkennen, ob Platine beziehungsweise Gerät USB 3.0 (Super Speed), 3.1 (Super Speed +), 3.2 (Super Speed 20 Gbps) oder gar Thunderbolt unterstützt.

Auch das Datenblatt verrät nicht immer, um welche Generation von USB 3 es sich handelt. Was also tun? Es gibt eine Reihe Symbole, die neben der Geschwindigkeit auch auf die Funktionen Power Delivery oder Displayport hinweisen.

Von USB 3.0 ist bereits das Doppel-S bekannt, das dem USB-Symbol neben dem Port anhängt. Handelt es sich um einen USB-3.1-Anschluss, der die 10 GBit pro Sekunde unterstützt, weist zusätzlich eine kleine, hochgestellte 10 am Symbol darauf hin. Manchmal ist im Netz auch ein kleines Plus am Doppel-S zu sehen, was auf das Gleiche hinweisen soll. Werden außerdem noch 100 Watt an Leistung geboten, prangt das Symbol auf einer kleinen Batterie. Der Displayport wird zusätzlich mit einem stilisierten „DP“ gekennzeichnet. Alles in allem hält das USB-Universum nun mehr als zehn Logos bereit. Eine Übersicht finden in der Randspalte rechts.

Handelt es sich bei der Typ-C-Buchse um Thunderbolt 3, ist sie meist mit einem kleinen Blitz gekennzeichnet. Der Port bietet dann den vollen Strauß aller Funktionen.

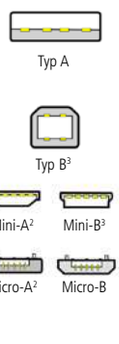
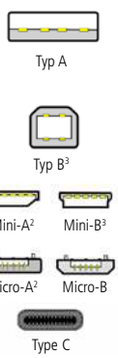
## Praxistest

In der Produktpraxis sind aber weder die Symbole einheitlich

# Wie USB-Schnittstellen und Stecker kombiniert werden

**Die erste USB-Variante verfügte noch über zwei Geschwindigkeiten namens Low- und Full-Speed, welche 1,5 beziehungsweise 12 MBit pro Sekunde übertragen konnten – das sind heute kaum mehr vorstellbare 0,2 bis 1,5 MByte/s!**

**Ab dem Frühjahr 2000 stand dann die finale Version von USB 2.0 mit dem Hi-Speed-Modus zur Verfügung. Mit bis zu 480 MBit pro Sekunde vervierzigfachte man die Übertragungsrate, von der im Praxiseinsatz allerdings weniger als 40 Megabyte pro Sekunde übrig bleiben. Da diese Übertragungsrate für moderne Festplatten immer noch viel zu langsam war, wurde 2008 die nächste Ausbaustufe USB 3.0 spezifiziert.**

| Alte/gebräuchliche (vorherige) Bezeichnung | USB 1.0/1.1 | USB 2.0 | USB 3.0 | USB 3.1 | USB 3.2 | Thunderbolt 3 (USB 4.0[1]) |
|---|---|---|---|---|---|---|
| Logo | | | | | | |
| Symbol | | | | | | |
| Max. Brutto-Datenrate | 1,5/12 MBit/s | 480 Mbit/s | 5 Gbit/s | 10 Gbit/s | 20 Gbit/s | 40 Gbit/s |
| Max. nutzbare Netto-Datenrate | 130 KB/s/1 MB/s | 40 MB/s | 300 MB/s | 900 MB/s | 1.800 MB/s | 3.600 MB/s |
| Neue/offizielle Bezeichnung | USB 1.0/1.1 | USB 2.0 | USB 3.2 Gen. 1 (ehem. 3.1 Gen. 1) | USB 3.2 Gen. 2 (ehem. 3.1 Gen. 2) | USB 3.2 Gen 2×2 | Thunderbolt 3 |
| Speed-Bezeichnung | Low Speed/Full Speed | High Speed | Super Speed | Super Speed+ | Super Speed++ | - |
| Max. Leistung (=V*A) | 0,5 Watt | 2,5 Watt | 4,5 Watt | 4,5 Watt[4] | 4,5 Watt4 | 10, 60 oder 100 Watt |
| Spannung | 5 V | 5 V | 5 V | 5 V | 5 V | 5, 12 oder 20 V |
| Stromstärke | 0,1 A | 0,5 A | 0,9 A | 0,9 A | 0,9 A | 2 oder 5 A |
| Anschlüsse |  Typ A, Typ B[3], Mini-A[2], Mini-B[3], Micro-A[2], Micro-B |  Typ A, Typ B[3], Mini-A[2], Mini-B[3], Micro-A[2], Micro-B, Type C | Typ A 3.0, Typ B 3.0, Micro-B 3.0[3], Type C | Type A, Type C | Type C | Type C |

[1] Noch nicht spezifiziert, wird im Sommer 2019 finalisiert.
[2] Kaum verwendet, sehr exotisch.
[3] Wird von USB-C ersetzt und stirbt aus.
[4] Ohne USB Power Delivery. Kann über USB-PD verfügen.

gestaltet, noch vollständig ausgewiesen. Der USB-3.0-Anschluss an Notebooks oder Mainboards ist beispielsweise nicht unbedingt blau. Der Hersteller färbt ihn gerne um, so wie es ihm ins Farbschema passt. Bei der USB-C-Buchse handelt es sich selten um das bisher lizenzpflichtige Thunderbolt 3, meist handelt es sich um USB 3.0 oder 3.1. Mit Glück ist auch bei Notebooks manchmal Displayport dabei. Im Zweifel hilft nur ein Blick ins Handbuch des Notebooks oder des Mainboards – oder auch schlichtes Ausprobieren.

## USB 3.1 Gen1 und Gen2

Bei Händlern kursieren darüber hinaus weitere Bezeichnungen. USB 3.0 wurde zwischenzeitlich auch als USB 3.1 Gen1 bezeichnet – und da das einfach viel fortschrittlicher klingt, haben viele Hersteller ihre Produkte einfach umgelabelt. Man merkt sich am besten einfach: Wo USB 3.1 Gen1 draufsteht, sind nur die altbekannten 5 GBit pro Sekunde möglich und es ist von USB 3.0 mit Super Speed die Rede.

Erst mit dem Zusatz „Gen2“ oder „Super Speed Plus“ bietet die USB-3.1-Schnittstelle die doppelte Geschwindigkeit des neueren Standards von 10 GBit pro Sekunde. Der gleiche Name – also USB 3.1 Gen1 und Gen2 – ergibt lediglich aus technischer Sicht Sinn: Die differenziellen Leitungspaare haben sich nicht verändert, nur der Takt hat sich verdoppelt und eine effizientere Kodierung (128b/132b statt 8b/10b) wird verwendet.

## Noch mehr Namen …

USB 3.2 setzte den vorläufigen Schlusspunkt. Und erneut kam es zu Änderungen bei den Bezeichnungen. Laut Spezifikation wird das ohnehin schon in USB 3.1 Gen. 1 umbenannte USB 3.0 nochmals in USB 3.2 Gen. 1 umbenannt – genauso wie USB 3.1 Gen. 2 in USB 3.2 Gen. 2. Um die Verwirrung perfekt zu machen, heißen die neuen 20 GBit/s nun USB 3.2 Gen. 2x2.

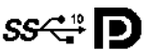
Um dem Chaos Herr zu werden, hilft es wie erwähnt, auf das Symbol zu blicken oder die korrekte „Speed“-Bezeichnung herauszufinden, etwa „Super Speed +“.

## Bandbreite-Überlegungen

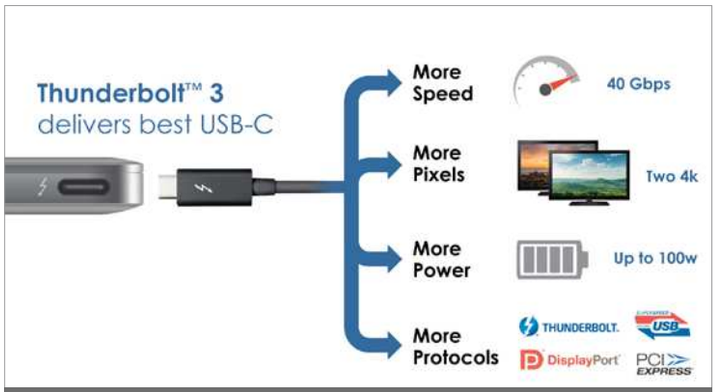
Beruhigend ist dabei zu wissen, dass Super Speed 20 Gbps (USB 3.2) mit nochmal doppelter 3.1-Geschwindigkeit von 20 GBit/s derzeit nicht

## Übersicht: Was die USB-Symbole bedeuten

**Bis zu zehn Symbolvarianten zeigen auf aktuellen USB-Geräten, was der Port kann. Leider halten sich nicht aller Hersteller daran und verwenden teils eigene „Interpretationen“ der offiziellen Symbole**

| | | **USB 2.0** High Speeds 480 MBit/s | **USB 3.0** (USB 3.1 Gen 1 oder USB 3.2 Gen 1) Super Speed 5 GBit/s | **USB 3.1 Gen 2 od. USB 3.2 Gen 2** Super Speed Plus 10 GBit/s |
|---|---|---|---|---|
| Ohne Power Delivery | Ohne Displayport | | | |
| | Mit Displayport | | |  |
| Mit Power Delivery | Ohne Displayport | | | |
| | Mit Displayport | | | |
| **Thunderbolt** mit Power Delivery, mit Displayport | | | | |

Und so sehen die Symbole in „freier Wildbahn“ aus: Der Thunderbolt-Blitz ist noch zweifelsfrei zu interpretieren, beim USB-Symbol in der Mitte übersieht man aber schon mal das Pluszeichen. Von Super Speed und Co. liest man zudem gar nichts. Immerhin geben die Farben einen Anhaltspunkt




Thunderbolt 3 kann viel, aber nicht alles: So reicht die Bandbreite in der Regel nicht aus, um externe Grafikkarten mit voller Geschwindigkeit zu befeuern

Der Teufel steckt manchmal im Detail: Der kleine Pfeil symbolisiert zwar Thunderbolt 3. Leider benötigt die Samsung-SSD eine ebensolche Gegenbuchse und ist nicht mit USB-Schnittstellen kompatibel

## Übersicht: USB-Power-Delivery-Profile

**Fünf Profile von zehn bis 100 Watt können in USB gewissermaßen eingebettet werden. Um höhere Ströme zu nutzen, müssen zusätzlich aktive Kabel verwendet werden**

| Profil | Anschlüsse | Spannung | Max. Strom | Max. Leistung | Endgeräte |
|---|---|---|---|---|---|
| 1 | Alle | 5 V | 2 A | 10 W | Standard-Profil, mobile Geräte, Smartphones, |
| 2 | Alle | 5 V | 2 A | 18 W | Tablets, kleine Notebooks, zukünftige Endgeräte |
| | USB-C | 12 V | 1,5 A | | |
| 3 | Alle | 5 V | 2 A | 36 W | Notebooks, größere Endgeräte |
| | USB-C | 12 V | 0,125 | | |
| 4 | Alle | 5 V | 2 A | 60 W | Notebooks, Hubs, Docking-Stations |
| | USB-C | 12 V | 3 A | | |
| | | 20 V | 3 A | | |
| 5 | Alle | 5 V | 2 A | 100 W | Workstations, Hubs, Docking-Stations |
| | USB-C | 12 V | 5 A | | |
| | | 20 V | 5 A | | |

unbedingt notwendig ist – zumindest noch nicht. Aktuelle USB-SSDs nutzen gerade mal die 3.0-Bandbreite von 5 GBit/s aus. Eine Sandisk Extreme 900 ist und bleibt seit vielen Jahren die Ausnahme, indem sie Datenraten um die 900 MByte/s schafft.

Die bessere Alternative sind externe SSDs, die Thunderbolt 3 nutzen und mit PCI-Express-SSDs vergleichbar sind, etwa eine Samsung SSD X5. Sie sind auch deutlich schneller als die USB-SSDs, haben aber auch Nachteile: Sie funktionieren nur an Thunderbolt-Buchsen, sind also nicht abwärtskompatibel zu USB-3.x-Buchsen, die häufiger zu finden sind. Außerdem sind sie noch sehr teuer, also noch teurer als die ohnehin schon teuren USB-SSDs.

Der Thunderbolt-Standard basiert technisch auf PCI Express. Damit steht nun zwar eine externe Verbindung mit guter Geschwindigkeit zur Verfügung, dieser sind aber auch Grenzen gesetzt. In der Regel ist Thunderbolt 3 mit lediglich vier PCIe-Lanes angebunden. Für externe Gaming-Grafikkarten zum Beispiel ist das zu wenig. Den im Gehäuse verbauten Pendants stehen 16 PCIe-Lanes zur Verfügung, was sich je nach Grafikkartenmodell und Spiel in circa 10 bis 30 Prozent mehr Leistung äußert.

### USB Power Delivery: bis zu 100 Watt

USB hat sich nicht zuletzt deswegen durchgesetzt, weil die Stromversorgung von Anfang an mitgekommen ist. So wurden etwa PS/2 und sämtliche seriellen und parallelen Schnittstellen abgelöst und alle mit dem gleichen Stecker angeschlossen. Größere Geräte, wie etwa ein Drucker, brauchten dennoch eine zusätzliche Stromversorgung, da die 2,5 Watt bei USB 2.0 oder auch die 4,5 Watt bei USB 3.0 zu wenig sind.

Die eigenständige Spezifikation USB Power Delivery (kurz: USB PD) beschreibt insgesamt fünf Profile mit 10 bis 100 Watt, die die PC-Peripherie mit der jeweils richtigen Spannung versorgt. Im Gegensatz zur herkömmlichen USB-Verbindung, die auch in der 3.1-Version noch immer nur 5 V bietet, versorgt USB PD Geräte auch mit Spannungen über 12 V und 20 V.

Für höhere Ströme und Spannungen sind aber aktive Kabel notwendig, die an den Steckern mit elektronischen Komponenten versehen sind. Am passiven Typ-A-Stecker gehen maximal 900 Milliampere und 4,5 Watt vorbei, am passiven Typ-C bis zu 3 Ampere und 15 Watt. Ein aktiver C-Stecker aber schafft bis zu 5 Ampere, sodass mit einem aktiven Kabel bis zu 25 Watt aus einem Port gezogen werden können, ohne USB PD zu verwenden. Bei Notebooks sollten stets die mitgelieferten Kabel verwendet werden.

### USB 4

Wie geht es weiter mit der USB-Schnittstelle? Das Wirrwarr könnte sich bald auflösen – mit der Betonung auf „könnte“; Intel hat mittlerweile die Thunderbolt-Protokollspezifikation dem USB-IF (Implementers Forum) kostenlos überlassen, also der für USB zuständigen Organisation. Das Wissen daraus wird als Basis für USB 4.0 dienen – wie die Spezifikation aber genau aussehen wird, ist noch offen. Bis zum Sommer 2019 soll sie laut USB-IF fertig sein.

USB 4.0 könnte so zumindest theoretisch alle alten Standards ablösen, der USB-C-Anschluss alle alten Steckervarianten ersetzen. Aus Anwendersicht wäre das mehr als begüßenswert. Jetzt, wo Thunderbird 3 ja nichts mehr kostet, steht dem Siegeszug hoffentlich nichts mehr im Wege.

# So funktioniert ein Mainboard: SATA & M.2

Serial-ATA oder kurz SATA ist eigentlich eine aussterbende Schnittstelle. Lange Jahre dominierte sie allerdings die Mainboards und viele Techniken beziehungsweise Fachbegriffe treten auch heute noch regelmäßig in Erscheinung. Grund genug für einen kleinen Blick zurück.

## Serial ATA

Vom Jahr 2001 an brachte Serial ATA nicht nur deutlich höhere Transferraten, sondern auch neue, einfacher zu handhabende Kabel. In der ersten Variante konnten bei einem Takt von 1,5 GHz annähernd 150 Megabyte nutzbare Daten pro Sekunde übertragen werden – seriell und nicht mehr parallel wie bis dato üblich. Für platterbasierte Festplatten reichte die Geschwindigkeit der ursprünglichen SATA-Spezifikation aus, denn damals dachte noch niemand an SSDs, die schlicht noch nicht erfunden waren. Doch mit leistungsfähigen Flash-Laufwerken wurden auch immer schnellere Versionen von SATA nötig.

Mit der zweiten Revision SATA 3 GBit/s, oft auch als SATA II bezeichnet, verdoppelte man die Übertragungsrate auf nutzbare 300 Megabyte pro Sekunde (inkl. 8b/10b-Kodierung: 3.000 MBit/s) und damit auf das Niveau der damals schnellsten SSDs.

## NCQ und AHCI

Hinzu kamen außerdem Funktionen wie AHCI („Advanced Host Controller Interface“) und NCQ („Native Command Queuing“). AHCI sorgte ab etwa 2004 für eine einfachere und einheitliche Treiberintegration. NCQ ordnet Daten innerhalb der Ströme so um, dass diese effizienter abgearbeitet werden können (siehe auch Grafik im Artikel „Festplatten“). Auch SSDs können von NCQ profitieren, da der Controller mehrere Zugriffe auf den Inhalt einer Flash-Speicherzelle kombinieren kann statt mehrfach die umständliche Read-Modify-Write-Prozedur für jeden Datenblock auszuführen.

Eine weitere interessante Technik ist „Staggered Spinup“ – diese Technik erlaubt das verzögerte Einschalten der Festplatten. Der Hintergrund: Das Anlaufen und Beschleunigen des Festplattenmotors sorgt beim Einschalten für eine sehr hohe Leistungsaufnahme; werden mehrere Festplatten gleichzeitig eingeschaltet, kann das zur Überlastung des Netzteils führen.

## SATA 6 GBit/s

Die dritte SATA-Generation bietet eine maximale Bandbreite von 600 MByte (nutzbare Bandbreite) oder 6.000 Megabit pro Sekunde (Bandbreite inklusive Prüfbits) und kann somit Daten doppelt so schnell übertragen wie der SATA-3-Gb-Standard (3.000 MBit/s). Daher wird diese auch oft „SATA 6 GBit/s“ genannt. Theoretisch lassen sich die gleichen Kabel weiterhin verwenden. Das SATA-Gremium gibt jedoch an, dass manche Kabel aus minderwertigem Material hergestellt sind und daher bereits mit SATA 3 GBit/s an der Leistungsgrenze liegen – hier lohnt sich ein Blick, wenn ältere Rechner Probleme machen.

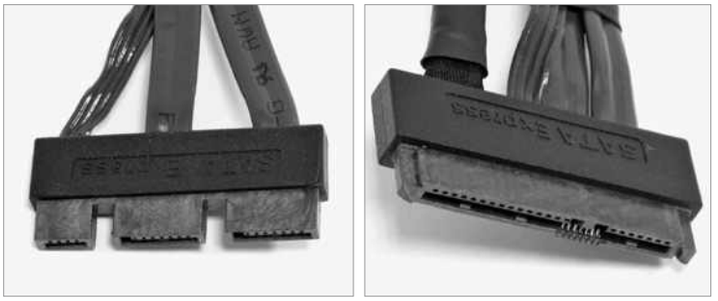
## Neue Anschlüsse

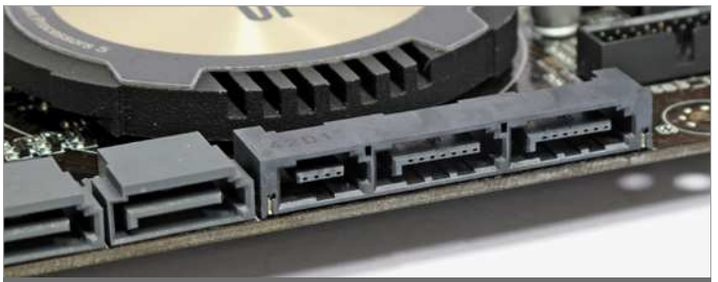
Zudem gibt es bei der dritten SATA-Generation zusätzlich kleinere Anschlüsse etwa für Notebooks, Netbooks oder DVD-Player. Während SATA-3-GBit/s-Geräte die Funktion Native Command Queuing (kurz: NCQ) unterstützen, bietet

Aktuelle SATA-Geräte basieren in der Regel auf dem SATA-6Gb/s-Standard – der höhere Ansprüche an die Kabel stellt. Dennoch sind wenige Probleme bekannt, die auf ältere SATA-3Gb/s-Kabel zurückzuführen wären. Mit PCI-Express und M.2 hat sich diese Frage ohnehin erledigt (siehe Fließtext)

## Theoretische Bandbreiten

| | Theoretische Bandbreite in Megabyte/s | Theoretische Bandbreite in Megabit/s |
|---|---|---|
| **USB 2.0** | 60 MByte/s | 480 MBit/s |
| **USB 3.0** | 625 MByte/s | 5.000 MBit/s |
| **USB 3.1** | 1.250 MByte/s | 10.000 MBit/s |
| **USB 3.2** | 2.500 MByte/s | 20.000 MBit/s |
| **SATA** | 187,5 MByte/s | 1.500 MBit/s |
| **SATA 3 GBit** | 375 MByte/s | 3.000 MBit/s |
| **SATA 6 GBit** | 750 MByte/s | 6.000 MBit/s |
| **SATA Express** | 2.000 MByte/s | 16.000 MBit/s |
| **Eine PCI-E-Lane** | 250 MByte/s | 2.000 MBit/s* |
| **Eine PCI-E-2.0-Lane** | 500 MByte/s | 4.000 MBit/s* |
| **Zwei PCI-E-2.0-Lanes** | 1.000 MByte/s | 8.000 MBit/s* |
| **Eine PCI-E-3.0-Lane** | 985 MByte/s | 7.880 MBit/s* |
| **Zwei PCI-E-3.0-Lanes** | 1.970 MByte/s | 15.760 MBit/s* |
| **LAN (100 MBit)** | 12,5 MByte/s | 100 MBit/s |
| **LAN (Gigabit)** | 125 MByte/s | 1.000 MBit/s |
| **Thunderbolt** | 1.250 MByte/s | 20.000 MBit/s |
| **Thunderbolt 2** | 2.500 MByte/s | 20.000 MBit/s |
| **Thunderbolt 3** | 5.000 MByte/s | 40.000 MBit/s |

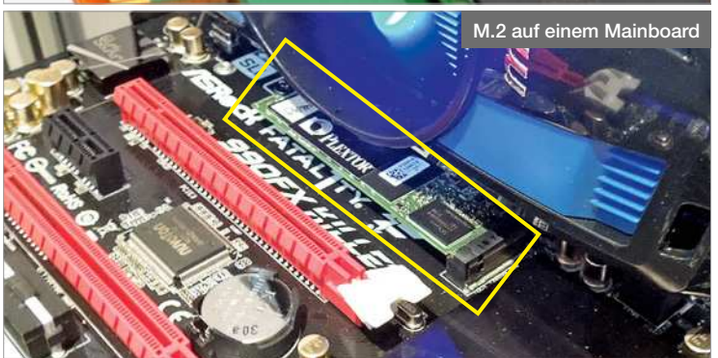




SATA Express fand sich zeitweise auf vielen Mainboards, ist mittlerweile aber nur noch selten anzutreffen. An einem solchen Anschluss lassen sich entweder ein SATA-Express-Laufwerk oder zwei „normale“ SATA-Laufwerke betreiben

M.2-Anschluss mit eingesteckter SSD

M.2 auf einem Mainboard



Der M.2-Anschluss zeichnet sich durch den geringen Platzverbrauch aus. In der Mitte sehen Sie eine SSD im Anschluss. Festgehalten wird diese durch eine einzelne Schraube am hinteren Ende.

SATA 6 GBit/s zusätzlich NCQ Streaming. Damit soll der Transfer von Audio- und Video-Dateien verbessert werden.

## Stromsparfunktionen

Neu waren auch mehrere Stromsparfunktionen. So musste die Festplatte bis dato auf Erlaubnis des Hosts warten, bevor sie in den Schlafzustand wechseln konnte. Bei SATA 6 GBit/s können sowohl Host als auch Laufwerk den Schlummer-Modus starten, ohne sich abstimmen zu müssen. Das Laufwerk muss auch nicht mehr vorher in den aktiven Modus gelangen.

## SATA-Bandbreite ausgeschöpft

Ab etwa 2013 nutzten viele SSD-Speicher bereits die volle Bandbreite von SATA 6 GBit/s aus – die Entwicklung hatte eine Dimension angenommen, welche die SATA-Entwickler so nicht vorhergesehen hatten. Die Technik-Roadmap des Konsortiums wurde buchstäblich überholt. Anfang 2014 mussten die Hersteller einsehen, dass die Entwicklung eines neuen SATA-Standards mit 12 GBit/s – bisher wurde die Geschwindigkeit der vorherigen Generation immer verdoppelt – bis zu zwei Jahre lang gedauert hätte. Das Ergebnis wäre zu spät marktreif gewesen.

Man entschied sich daher für eine preiswertere und energiesparende, gewissermaßen improvisierte Alternative. Statt am alten Standard festzuhalten, bediente sich das Entwicklerkonsortium des etablierten und ausgereiften PCI-Express-3.0-Standards, welcher pro Lane auf ein GB/s kommt (siehe PCI-Express-Artikel). Heraus kam eine Mischform, die sowohl mit SATA- als auch mit PCI-Express-Anschlüssen funktioniert und vom Konsortium entsprechend vermischt „SATA Express“ genannt wurde.

## SATA Express

SATA Express verwendet zwei PCI-Express-2.0-Lanes und erreicht somit annähernd ein GByte/s. Zur physischen Verbindung werden die bekannten SATA-Buchsen verwendet, jeder SATA-Express-Anschluss besitzt zwei davon – zusätzlich aber noch eine dritte!

Ältere SATA-Laufwerke schließt man einfach an einen der beiden Ports an – so ist die Abwärtskompatibilitätgewährleistet. Die effektive Geschwindigkeit liegt dann bei den für konventionelles SATA 6 GBit/s maximal spezifizierten 600 MByte/s.

SATA-Express-Laufwerke dagegen belegen nicht nur die beiden „herkömmlichen“ SATA-Anschlüsse, sondern zusätzlich auch noch die dritte Buchse. Ein einzelnes Laufwerk bringt somit eine ganze Steckerleiste mit.

## Neue Lösungen

Nimmt man die schnellsten Ein-Controller-SSDs (ca. 1.500 MByte/s) zum Zeitpunkt der Markteinführung als Maßstab, war SATA Express bereits damals veraltet. Aber auch mit zeitgemäßen Chipsätzen, bei denen SATA Express mit PCI 3.0 angebunden wurde, verdoppelte sich die Bandbreite auf „nur“ knapp 2.000 MByte/s. Das war für eine Intel SSD 750 und eine Samsung SSD 950 Pro immer noch zu langsam.

Warum aber überhaupt SATA-Komponenten nutzen, wenn der Transport der Daten ohnehin über PCI-Express erfolgt? Schon als es noch keine nativen SSD-Controller für PCI-E gab, nutzten viele Hersteller die Geschwindigkeit des Erweiterungsslots, um mit RAID-0-verschalteten SATA-SSDs auf einer Platine das SATA-Limit zu umgehen. Der Vorteil von PCI-E ist, dass einer SSD über diesen Weg die in der Pra-

xis bestmögliche Anbindung zur Verfügung steht.

## M.2 als PCI-Express-Vermittler

So kam die Lösung des Problems erst, als PCI-Express ohne SATA-Altlasten für die Verbindung von SSDs genutzt wurde. Die bis heute populärste Umsetzung stellt dabei der M.2-Formfaktor dar.

Eigentlich handelt es sich bei M.2 (ehemals NGFF, „Next Generation Form Factor“) nur um ein spezielles Steckformat, das mit verschiedenen Standards arbeitet: Es erlaubt den Einsatz von Steckkärtchen, die meist 22 mm breit und 42, 60, 80 oder 110 mm lang sind (siehe Kasten rechts unten) – in erster Linie für den Einsatz in Notebooks. Doch mittlerweile ist die Schnittstelle auch bei Mainboards so beliebt, dass kaum noch ein neues Modell ohne auskommt. Der Grund dafür ist wohl, dass sich M.2 günstig und Platz sparend unterbringen lässt – und natürlich, dass es bereits eine Reihe von M.2-Hardware gibt. Im Gegensatz zu normalen PCI-Express-Karten steht eine M.2-Erweiterungskarte meist nicht senkrecht zum Mainboard, sondern liegt eng an der Platine an. Hier merkt man die Ursprünge des Formfaktors bei den Ultrabooks

Dass PCI-Express im M.2-Format die alte SATA-Schnittstelle ablösen wird, daran gibt es mittlerweile wenig Zweifel; bis zu 4.000 MB/s statt 600 MByte/s theoretische Maximalleistung sind ein großer Sprung nach vorn. Ein weiterer Vorteil ist, dass sich auch M.2-SSDs per Adapter sehr einfach zu normalen PCI-E-Steckkarten umfunktionieren lassen. So besteht auch für ältere Systeme, die weder einen SATA-Express- noch einen M.2-Anschluss aufweisen, die Möglichkeit, schnelle SSDs betreiben zu können.

## M.2-Vielfalt

Aber M.2 ist mehr als nur eine High-End-SSD-Schnittstelle, M.2-SSDs sind nur die bekannteste Anwendung des Formfaktors. Möglich sind auch WLAN- und andere Erweiterungsmodule sowie gänzlich andere Schnittstellen wie USB 2.0, das vom Raspberry Pi oder Arduino bekannte I2C und nicht zuletzt „klassisches“ SATA. Was davon in einem M.2-Slot betrieben werden kann, hängt auch vom Host ab. Die meisten modernen Mainboards unterstützen mit M.2 auch SATA, die Standard-Anbindung erfolgt aber per PCI-Express.

## Kerben dechiffriert

Über die genaue Funktionalität geben Anzahl und Position der Kerben Aufschluss: WLAN-Module sind für Slots mit sogenannter E-Kodierung gedacht, SSDs setzen dagegen auf den sogenannten M-Key (siehe Übersicht rechts oben).

Mechanisch werden diese Kodierungen durch eine Kerbe in der Kontaktleiste der SSD und eine korrespondierende Nase im M.2-Steckplatz gekennzeichnet. Die Folge: M.2-SSDs (M-Kerbe) passen nicht in M.2-Steckplätze, die für WLAN-Module gedacht sind (E-Nase) – und umgekehrt. Als dritte M.2-Kodierung (von insgesamt 12 spezifizierten) findet sich auf einigen älteren Modellen übrigens noch der B-Key. Die Geschwindigkeit dieser alten Implementationen lässt sich nur mit WLAN-Modulen sinnvoll nutzen, die auch meist sowohl über eine E-, wie auch eine B-Kerbe verfügen.

Obwohl M.2 wohl die Laufwerks-Schnittstelle der Zukunft ist, werden SATA 6 GBit/s und SATA Express noch eine ganze Zeit lang auf dem Markt präsent sein – vor allem bei externen SSDs und Festplatten sind diese Varianten populär.

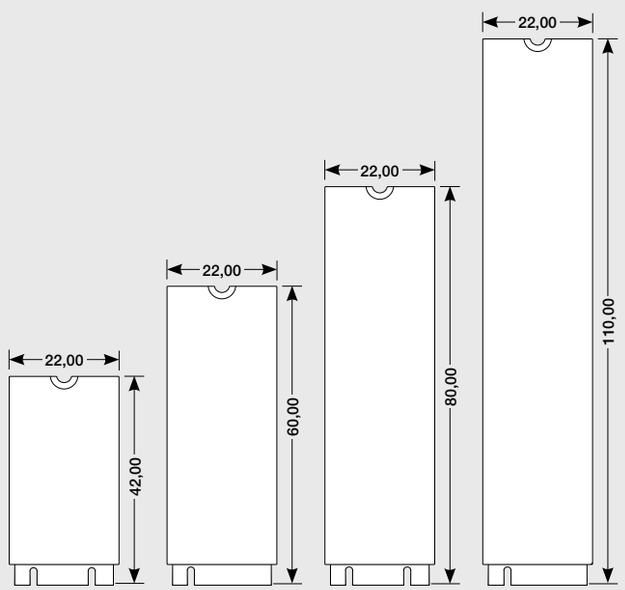
# Was die Kerben über M.2 verraten

| | Theoretische Bandbreite in Megabyte/s | Theoretische Bandbreite in Megabit/s |
|---|---|---|
| **A** | 8–15 | PCI-Express ×2, USB 2.0, I2C und DP ×4 |
| **B*** | 12–19 | PCI-Express ×2, SATA, USB 2.0 und 3.0, Audio, PCM, IUM, SSIC und I2C |
| **C** | 16–23 | Reserviert für zukünftigen Gebrauch |
| **D** | 20–27 | Reserviert für zukünftigen Gebrauch |
| **E** | 24–31 | PCI-Express ×2, USB 2.0, I2C, SDIO, UART und PCM |
| **F** | 28–35 | Future Memory Interface (FMI) |
| **G** | 39–46 | Allgemein (In der M.2-Spezifikation nicht vorgesehen) |
| **H** | 43–50 | Reserviert für zukünftigen Gebrauch |
| **J** | 47–54 | Reserviert für zukünftigen Gebrauch |
| **K** | 51–58 | Reserviert für zukünftigen Gebrauch |
| **L** | 55–62 | Reserviert für zukünftigen Gebrauch |
| **M*** | 59–66 | PCI-Express ×4 und SATA |

*für SSDs in Gebrauch befindliche Kerbungsarten

# Wie groß sind M.2-SSDs?

**Bei bisherigen Laufwerken war man entweder auf den 5,25 Zoll-, 3,5 Zoll- oder 2,5-Zoll-Formfaktor begrenzt. M.2 ist wesentlich flexibler. Eine M.2-Platine kann 16, 26, 30, 38, 42,60, 80, oder 110 mm lang sein. Die Breite darf 12, 16, 22 oder 30 mm betragen. Die SSD-Hersteller beschränken sich aktuell aber auf SSDs in einem der abgebildeten Formate.**



# So funktioniert ein Mainboard: Speicher (RAM)

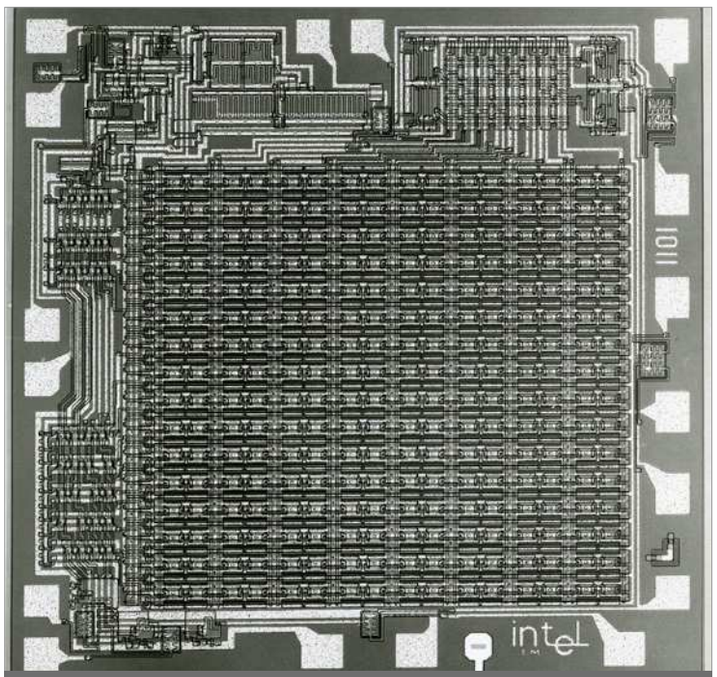
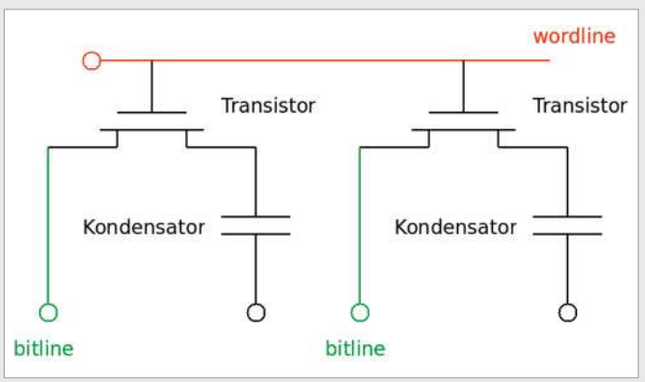



Bild: Intel

Speicher ist vergleichsweise simpel aufgebaut und wird wie die meisten Chips auf Wafer geätzt. Hier einer der ersten SRAM-Chips von Intel (Intel 1101). Die Ausbeute am Anfang: Zwei (statt wie angepeilt 20) Dies pro Wafer

## Wie DRAM Daten speichert

**Eine DRAM-Speicherzelle benötigt lediglich einen Transistor und einen Kondensator, an eine Wordline können theoretisch beliebig viele Zellen angeschlossen werden. Anders als eine SRAM-Speicherzelle verfügt eine DRAM-Zelle lediglich über einen Ausgang und arbeitet zudem deutlich langsamer.**




Der Arbeitsspeicher ist wichtiger Bestandteil eines jeden PCs , in vielen Fällen aber auch dessen größter Bremsklotz. Warum ist das so?

### Grundbegriffe

Der im Englischen für Arbeitsspeicher verwendete Begriff RAM steht für „Random Access Memory“. Mit diesem Namen wird das Prinzip hervorgehoben, dass jede beliebige, also zufällig gewählte Speicherzelle in exakt gleicher Zeit angesprochen und ausgelesen werden kann. Umgangssprachlich wird zwar häufig zwischen RAM und ROM („Read-only memory“, nur lesbarer Speicher) unterschieden, technisch handelt es sich bei den meisten ROM jedoch genauso um RAM.

Das gilt auch für Flash-Speicher, wie er in USB-Sticks oder SSDs zum Einsatz kommt: Auch hier kann jeder beliebige Block in konstanter Zeit ausgelesen werden.

### Speicher-Probleme

Das Problem, Daten zu speichern, ist schon knapp 100 Jahre alt. Bereits in den 1920er-Jahren machten sich Wissenschaftler Gedanken darüber, wie Werte elektronisch „festgehalten“ werden können. Nach den ersten Trommelspeichern in den 1930er-Jahren, einer Art Vorläufer der Festplatte, dauerte es aber noch bis 1946, ehe Daten auch in Speicherröhren abgelegt werden konnten. Diese waren jedoch sehr fehleranfällig und wurden bald durch Kernspeicher ersetzt, die immerhin schon einige Kilobyte speichern konnten.

„Richtigen“ Speicher, der die heute verwendeten Prinzipien aufgreift, gab es aber erst nach der Erfindung des Metalloxid-Halbleiter-Feldeffekttransistors (MOSFET) in den 1960er-Jahren.

### Die Geburt des RAM

Der Erfinder Robert Dennard hatte den Einfall für DRAM 1966 auf seiner Wohnzimmercouch und rief seinen Chef bei IBM noch am gleichen Tag um 22 Uhr an. „Ich wusste, dass es eine große Sache ist, aber ich wusste nicht, dass sie mal einen so großen Einfluss haben würde wie heute“, bekannte der Ingenieur einmal Jahrzehnte später. Veröffentlicht wurde das Patent 1968 – die moderne Arbeitsspeicherzelle war geboren.

Das von zwei ehemaligen Mitarbeitern von Fairchild Semiconductor gegründete Unternehmen Intel brachte im Jahr darauf den ersten Speicherchip mit MOSFETs und Silizium-Gates auf den Markt, den Intel 1101 mit einer Kapazität von 1 Kilobit. Dabei handelte es sich um SRAM, das „S“ steht hierbei für „static“. Dieser Speicher zeichnet sich dadurch aus, dass sein Inhalt erhalten bleibt, solange Spannung anliegt (mehr dazu im Prozessor-Teil).

### Flüchtige Speicher

Arbeitsspeicher ist allerdings aus DRAM-Speicherzellen aufgebaut. Namensgebend für diesen „dynamischen“ Speichertyp ist die Eigenschaft, dass die gespeicherte Information flüchtig ist. Das liegt daran, dass bei DRAM ein Bit nicht über ein aus sechs Transistoren bestehendes Flipflop gespeichert wird (siehe Prozessor-Teil), sondern durch die Kombination aus einem Kondensator (Speicher) und einem Transistor (zur Ansteuerung).

Wird die Speicherzelle adressiert und Strom angelegt, schaltet der

Transistor. Fließt die Ladung des Kondensators ab, lässt sich das als logische 1 interpretieren, das Ausbleiben entspricht einer 0. Über den gleichen Transistor lässt sich die Ladung auch in den Kondensator einspeisen.

Die Flüchtigkeit der Informationen ist dadurch bedingt, dass die Ladung auch ohne Schaltvorgang aufgrund von unerwünschten, bei höheren Temperaturen stärker ausgeprägten Leckströmen abfließt und die Ladung des Kondensators daher regelmäßig aufgefrischt werden muss. Bei modernen Speichermodulen ist eine Auffrischung aller Speicherzellen in Intervallen von wenigen ms vorgesehen.

## DRAM-Eigenheiten

Gibt es keine Auffrischung, verschwindet über kurz oder lang die Ladung und damit auch die Information. Ein Vorteil von DRAM ist dagegen die hohe Speicherdichte, da die benötigte Fläche pro Bit um ein Vielfaches geringer ist als bei vergleichbaren Speicherlösungen.

## Funktionsprinzip

Jede Speicherzelle eines DRAM-Moduls besteht wie erwähnt aus einem Kondensator und einem Transistor. Der Kondensator übernimmt das eigentliche Speichern der Information, der Transistor ist für Lese- und Schreibzugriffe zuständig. Jede Speicherzelle kann somit ein Bit speichern.

Für die Ein- und Ausgabe-Operationen verfügt jede Zelle zusätzlich über eine „Wordline“ und eine „Bitline“. Die „Wordline“ gibt den Befehl zur Schaltung an den Transistor, wodurch die Information über den Zustand des Kondensators (1 oder 0) an die „Bitline“ weitergeleitet wird. An eine „Wordline“ können theoretisch beliebig viele Speicherzellen angeschlossen werden. Eine derart verbundene Sammlung von Speicherzellen bezeichnet man als „Page“ oder auch Speicherzeile.

## Speicherfelder

Eine Speicherzeile ist bei aktuellen Systemen meist zwischen 512 und 2.048 Zellen „lang“. Die Anzahl der

# Wie ein Speichermodul aufgebaut ist

### Speicherchips

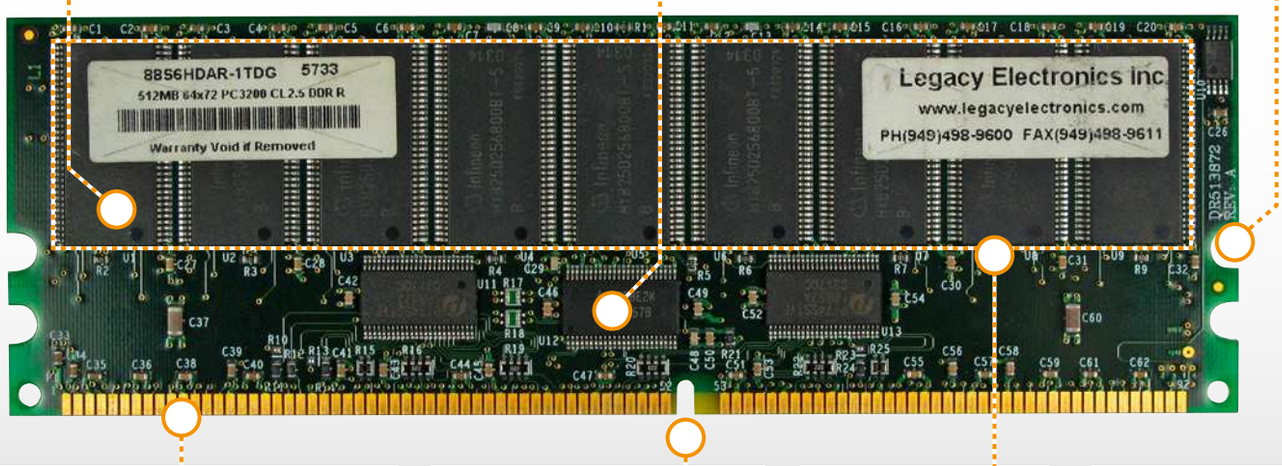
Je nach Kapazität des Speichermoduls enthält es unterschiedlich viele Speicherchips. Die meisten Module mit hoher Kapazität verfügen über Chips auf Vorder- und Rückseite.

### Register

Diese zusätzlichen Chips finden sich ausschließlich bei Registered-Modulen (Server). Sie entlasten den Speichercontroller und ermöglichen höhere Kapazitäten.

### Kerbe

Die Kerben an der Seite fixieren die Module in den Speichersockeln und verhindern ein ungewolltes Herausrutschen. SO-DIMM-Steckplätze verfügen aufgrund der höheren Anforderungen zusätzlich über Metallsicherungen.




### Pins

Auch die Anzahl der Kontakte variiert – 168 bei SDR-SDRAM, 184 bei DDR-SDRAM, 240 bei DDR2- und DDR3-SDRAM, 288 bei DDR4-SDRAM. SO-DIMM-Module, die speziell für Laptops entwickelt wurden, verfügen über 144 (SDR), 200 (DDR, DDR2), 204 (DDR3) oder 260 (DDR4) Kontakte.

### Kerbe

Da die Spannung von SDR, DDR-, DDR2-, DDR3- und DDR4-Modulen unterschiedlich ist, verfügen sie über unterschiedliche Kerben. Dadurch wird sichergestellt, dass sie nur in den richtigen Slot passen.

### Rank

Eine logische Speichereinheit wird als „Rank“ bezeichnet. Sie muss immer 64 Bit breit sein und besteht aus 4, 8 oder 16 Chips.

### ECC

ECC-Module verfügen je Byte über ein Paritätsbit und somit bei gleicher Kapazität über zusätzliche Speicherchips.

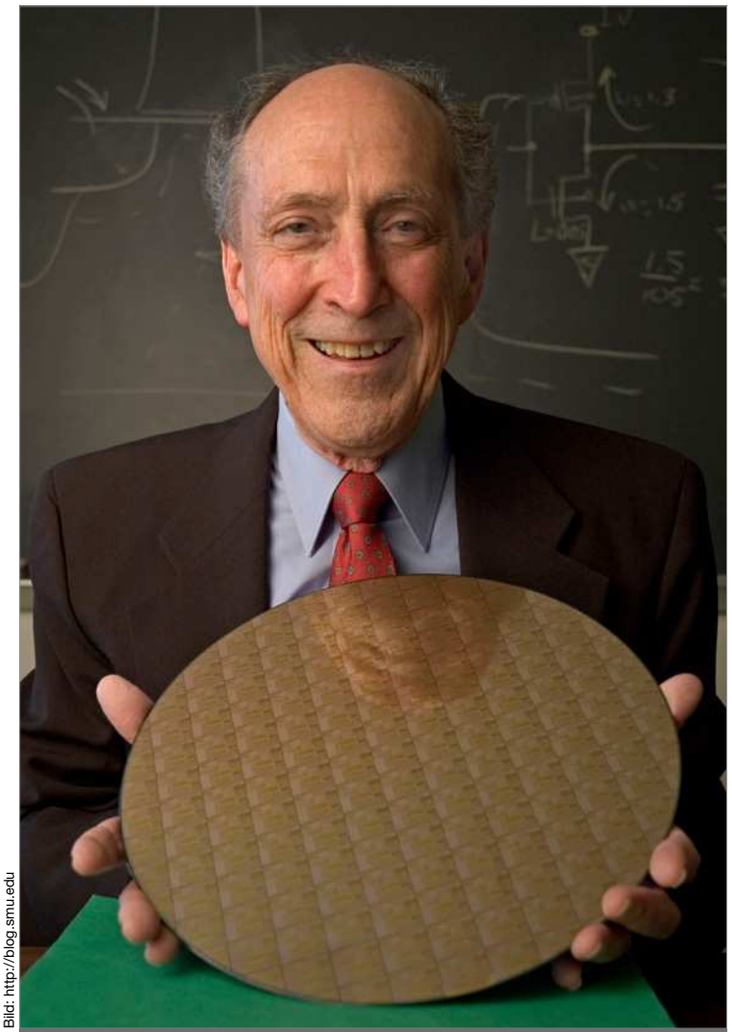
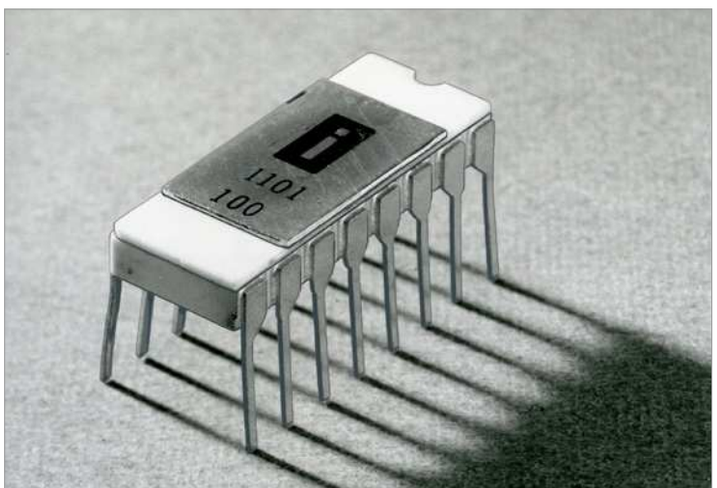


Bild: http://blog.smu.edu

Robert Dennard erhielt für die Erfindung von DRAM im Jahr 1966 zahlreiche Auszeichnungen




Dennards Erfindung führte unter anderem zur Gründung von Intel, wo man bald die ersten DRAM-Chips produzierte (hier der 1101, dessen Die-Shot auf der vorhergehenden Doppelseite zu sehen ist

Zeilen wiederum variiert – sie liegt bei den meisten aktuellen Chips in einer Größenordnung von 4.096 bis 16.384.

Man kann sich also einen Speicherchip als Matrix aus – beispielhaft – 16.384 Zeilen und 1.024 Spalten vorstellen; in der Summe könnte man in einem solchen Speicherfeld 16 Megabit oder 2 Megabyte speichern (siehe Abbildung auf der folgenden Doppelseite; mittlerweile sind die Module deutlich komplexer, der Einfachheit halber wählen wir aber dieses ältere Modul als Beispiel).

## Byteweiser Zugriff

Nun arbeiten x86-Prozessoren aber nicht mit einzelnen Bits, sondern lesen Daten immer byteweise aus. Intern ist ein Speicherfeld daher so organisiert, dass der Zugriff auf einen Punkt in der Matrix nicht ein, sondern gleich acht Bits zurückliefert. Vereinfacht kann man sich vorstellen, dass jedes Speicherfeld achtmal existiert. Wir bilden dies in der Grafik dadurch ab, dass unser Speicherfeld achtmal eingezeichnet ist. Die Page-Größe läge in diesem Fall daher auch bei 8 KiBit. Zusammen verfügen die acht Speicherfelder über 134.217.728 einzelne Zellen (16.384 Zeilen x 1.024 Spalten x 8 Bits). Sie sind mit dieser Organisation in der Lage, 128 Megabit oder 16 Megabyte an Informationen aufzunehmen.

## Speicherbänke

Ein solches Konglomerat aus Speicherfeldern nennt man „Bank“. Damit möglichst viele Daten parallel ausgelesen werden können (ein einzelner Zugriff dauert vergleichsweise lange, dazu gleich mehr), stecken in jedem Speicherchip mehrere Speicherbänke. Meist sind es ebenfalls genau acht.

Das Besondere daran ist, dass sich alle Bänke unabhängig voneinander ansprechen lassen – so muss nicht jedesmal gewartet werden, bis ein langwieriger Speichervorgang abschlossen ist.

Die Kapazität eines solchen Chips mit acht Bänken betrüge nun schon 1.024 Megabit oder 128 Megabyte. Weitere Steigerungen lassen sich ab einem gewissen Punkt nur noch mit dem Fertigungsverfahren erreichen, da die Chipgröße nicht beliebig wachsen kann. Mehr Kapazität erhält man aber vergleichsweise einfach durch die Kombination mehrerer Speicherchips.

## Speicher-„Chips“ und „Ranks“

Um eine höhere Flexibilität bei der Fertigung und eine noch größere Parallelität zu erreichen, werden mehrere Speicherchips schließlich zu einem „Rank“ zusammengeschlossen. Man kann sich einen Rank in etwa als unabhängiges Speichermodul auf einem Riegel vorstellen.

Ein aktueller Speicherriegel beherbergt typischerweise 8 oder 16 einzelne Speicherchips. Erneut wird so die Gesamtkapazität gesteigert, in unserem Beispiel auf nunmehr 16.384 Megabit oder 2.048 Megabyte oder 2 Gigabyte, verteilt auf 16 Chips, je 8 auf der Vorder- und 8 auf der Rückseite des Moduls.

## Bitbreiten

Dabei ist es wichtig, dass die aggregierte Bandbreite einer Speichereinheit zu der Busbreite des PCs passt – bei aktuellen PCs sind dies seit SDRAM-Einführung immer 64 Bit. Jeder Rank muss daher 64-bittig angebunden sein, und das sorgt oft für weitere Verwirrung.

Es hängt nämlich von der Zahl und Art der Speicherchips ab, wie der Speicherhersteller seine Ranks auf einem Modul anordnet. Ein paar

Beispiele sollen das illustrieren: Das rechts oben auf dieser Seite abgebildete Speichermodul verfügt über Chips, deren Bänke intern acht Felder haben (Vorsicht: Nicht verwechseln mit der Zahl der Bänke, die ebenfalls bei acht liegt). Es sind daher acht solcher Chips nötig, um auf 64 Bit zu kommen und einen Rank zu füllen. Die Zahl der Felder wird manchmal auch als Breite oder Länge eines Chips bezeichnet.

## Interne Organisation

Es gibt aber auch Speicherchips, deren Felder nur vier Bits gleichzeitig liefern – hier wären dann 16 Chips nötig, um auf die 64 Bit pro Takt zu kommen. Und damit nicht genug: Auch Speicherfelder mit intern 16 Matrizen sind möglich – es genügen dann vier Chips, um einen Rank zu bilden.

Mittlerweile sind auch schon Quad-Rank-Speichermodule verfügbar; diese müssen immer mit x4-Chips zusammengestellt sein, da sonst mehr als 16 Chips auf dem Modul untergebracht werden müssten. Das ist zwar theoretisch möglich, wird im Consumer-Bereich aber nicht gemacht.

## Packungsdichte

Doch warum ist das überhaupt wichtig? Der Grund liegt in den physikalischen Gegebenheiten, den so genannten kapazitiven Lasten. Ein Modul mit geringer Feldbreite und vielen Chips verursacht eine höhere kapazitive Last für die Speicher-Verbindung, was das System anfälliger macht für Signalrauschen. Das hat zur Folge, dass unter Umständen die Taktfrequenz reduziert oder die Latenzzeiten verlängert werden müssen, um den Speicher noch zuverlässig auslesen zu können.

Wer mehr als ein Modul je Speicher-Kanal verbauen will, ist hier mit single-ranked Modulen immer besser dran, die allerdings in der Regel bei gleicher Größe und gleichen Timings teurer sind als double-sided Module. Ausnahmen bestätigen aber die Regel: So profitieren einige Prozessoren von Dual-Rank-Modulen; der Effekt beruht darauf, dass Dual-Rank-Technik Rank-Interleaving ermöglicht.

## Single vs. Double sided

Oft liest man bei der Produktbeschreibung eines Speichers die Angaben „double-sided“ oder „single-sided“. Entgegen der landläufigen Meinung haben diese Bezeichnungen nichts mit den „Ranks“ zu tun. Hier geht es einzig und alleine darum, ob Chips nur auf einer Seite oder auf beiden angebracht sind – eine Angabe, die den meisten mit dem Wissen um die Rank-Struktur egal sein dürfte. Es gibt demnach Double-Rank-Module, die nur auf einer Modulseite mit Chips bestückt sind und Single-Rank-Module, die beidseitig bestückt sind.

## Lese- und Schreibzugriffe

Die Organisationsstruktur des Arbeitsspeicher ist schon eine Wissenschaft für sich ... Aber wie kommen die Daten nun vom Speicher in den Prozessor?

Grundsätzlich ist ein Lese-/Schreibzugriff in drei Phasen unterteilt: Vorbereitung, Ausführung des eigentlichen Lese-/Schreibvorgangs in einer Speicherzeile und Nachbereitung.

Während der Vorbereitung werden zwei Schritte durchgeführt: Zunächst ermittelt der Adressdekoder aus der ihm übergebenen Zeilenadresse die korrekte Speicherzeile. Die Länge dieser Adresse hängt dabei von der Anzahl der in der Speicherbank vorhandenen Speicherzeilen ab – bei einer 14 Bit

Derselbe Riegel in der Gesamtansicht: Es handelt sich um einen einseitig bestückten („Single sided“) Riegel mit einem „Rank“ (acht Chips sind mit jeweils acht Bit an den Controller angebunden)

Die Organisationsstruktur ist häufig auf einem Aufkleber vermerkt: Dieser Riegel arbeitet mit Chips, die jeweils acht Gigabit aufnehmen können und achtbittig angebunden sind

## Wie Sie RAM-Werte ermitteln

**Wenn der Speicher schon verbaut ist, helfen Tools wie der „Thaiphoon Burner“ weiter.**

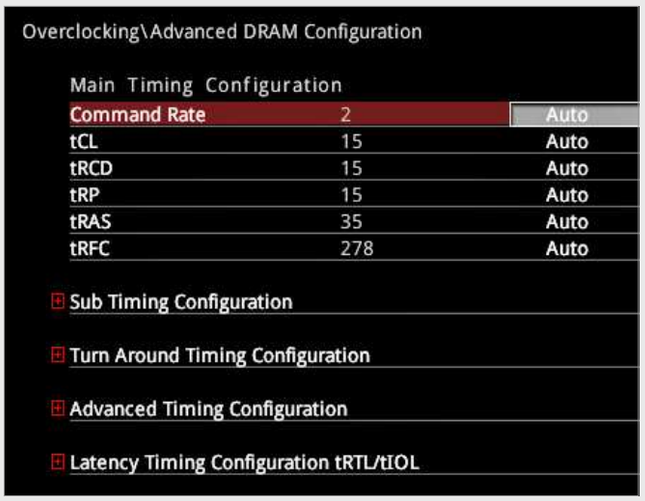
**Klicken Sie im Hauptmenü die Schaltfläche „Read“ an und wählen Sie das gewünschte Modul aus. Wenige Sekunden später erhalten Sie die Informationen über den Hersteller, die Produktnummer, Revision und Strukturbreite. Die Software liefert Ihnen unter anderem Informationen zu**

1. dem Hersteller der verbauten Speicherchips
2. der Produktnummer der Bausteine (Tipp: Via Online-Suche finden Sie bei Bedarf das Datenblatt!)
3. der Chip-Revision und der Strukturgröße und
4. der Anzahl der Ranks.

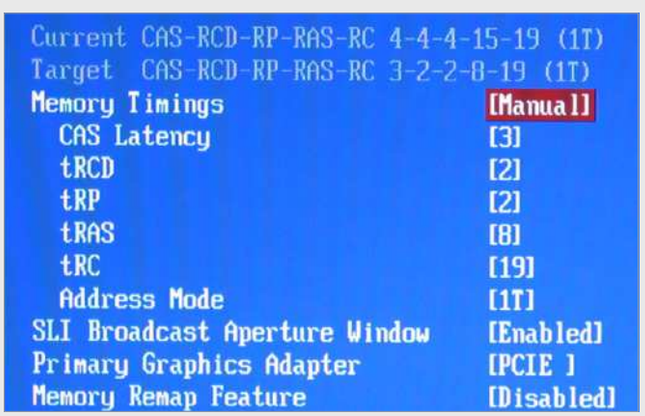
| MEMORY MODULE | | DRAM COMPONENTS |
|---|---|---|
| MANUFACTURER G.Skill | 1 | MANUFACTURER Hynix |
| SERIES Trident Z RGB | 2 | PART NUMBER H5AN8G8NAFR-TFC |
| PART NUMBER F4-3200C16-8GTZR | | PACKAGE Standard Monolithic 78-ball FBGA |
| SERIAL NUMBER 00000000h | 3 | DIE DENSITY / COUNT 8 Gb A-die (21 nm) / 1 die |
| JEDEC DIMM LABEL 8GB 1Rx8 PC4-2133-UA1-11 | | COMPOSITION 1024M x8 (64M x8 x 16 banks) |
| ARCHITECTURE DDR4 SDRAM UDIMM | | CLOCK FREQUENCY 1067 MHz (0,938 ns) |
| SPEED GRADE DDR4-2133 | | MINIMUM TIMING DELAYS 15-15-15-36-50 |
| CAPACITY 8 GB (8 components) | | READ LATENCIES SUPPORTED 16T, 15T, 14T, 13T, 12T, 11T, 10T |
| ORGANIZATION 1024M x64 (1 rank) | 4 | SUPPLY VOLTAGE 1,20 V |
| REGISTER MODEL N/A | | XMP CERTIFIED 1600 MHz / 16-18-18-38-56 / 1,35 V |
| MANUFACTURING DATE Undefined | | XMP EXTREME Not programmed |
| MANUFACTURING LOCATION Taipei, Taiwan | | SPD REVISION 1.1 / September 2015 |
| REVISION / RAW CARD 0000h / A1 (8 layers) | | XMP REVISION 2.0 / December 2013 |

## Die wichtigsten Optionen auf einen Blick

Die „Hauptlatenzen“ CAS, RCD, RP und RAS sind in der Regel über das UEFI-Setup einsehbar und können dort auch (in Grenzen) geändert werden. Nachfolgend eine Übersicht über die gängigsten Parameter:




- **CAS Latency (CL)**
  Wartezeit nach einem Spaltenzugriff bis zum Burst
- **RAS to CAS Delay (tRCD)**
  Wartezeit beim Umschalten von Zeilen- auf Spaltenzugriff
- **RAS to Precharge (tRP)**
  Wartezeit beim Anlegen der Referenzspannung auf eine Zeile
- **Active to Precharge (tRAS)**
  Wartezeit vor dem Zurückschreiben einer Zeile in den Speicher
- **Fast RW Turn around**
  Wartezeit zwischen Lese- und Schreibvorgang
- **DRAM Command Rate (CMD)**
  Wartezeit bei der Auswahl der korrekten Speicherchips
- **Bank-Interleaving**
  Wechselseitiger Zugriff auf die internen Bänke eines Speicherchips




langen Zeilenadresse können 16.384 unterschiedliche Zeilen adressiert werden. Anschließend wird die benötigte Speicherzeile aktiviert, indem zunächst die im Ruhezustand an der „Bitline“ anliegende Spannung gekappt und anschließend an die „Wordline“ Spannung angelegt wird.

Im zweiten Schritt wird nun der eigentliche Lese- oder Schreibzugriff auf die Speicherzeile durchgeführt. Bei einem Lesezugriff wird zunächst die Spaltenadresse ermittelt und die benötigten „Bitlines“ über einen Leseverstärker, der aufgrund der Länge der Leitungen notwendig ist, mit der Datenleitung des Speicherchips verbunden. Diese Daten werden dann mit dem nächsten Taktsignal ausgegeben.

### Schreibvorgänge

Ein Schreibvorgang verläuft ähnlich. Zunächst wird die Spaltenadresse vom Spaltendekoder entschlüsselt und die benötigten Speicherzellen mit der Datenleitung verbunden. Die zu schreibenden Informationen laufen nun durch den Leseverstärker und setzen die Bits in den korrespondierenden Speicherzellen.

Im Nachgang an einen Lese-/Schreibzugriff muss die Speicherzeile wieder deaktiviert werden. Hierfür wird die „Wordline“ wieder von der anliegenden Spannung getrennt, wodurch die Transistoren in den jeweiligen Speicherzellen dieser Speicherzeile die Kondensatoren der Speicherzellen von den Bitleitungen abkoppeln. Zuletzt werden die „Bitlines“ wieder mit Spannung versorgt und nehmen ihren Ursprungszustand an.

### Auffrischung

Wie bereits erwähnt unterscheidet sich dynamisches RAM von statischem RAM dadurch, dass die Daten flüchtig sind und regelmäßig aufgefrischt werden müssen. Diese Wiederauffrischung führen die primären Leseverstärker in definierten Abständen (meist 32 oder 64 Millisekunden) durch.

Dabei macht man sich die Tatsache zunutze, dass die Daten bei jedem Schreibvorgang die Leseverstärker durchlaufen und erst dann in die Speicherzellen geschrieben werden. Die Leseverstärker sind mit T4-SRAM, also 1-Bit-Datenspeicher als Flipflop realisiert, mit vier Transistoren ausgestattet und somit in der Lage, jeden beliebigen Zustand so lange zu halten, wie Spannung anliegt. Der Refresh erfolgt also regelmäßig aus diesen „Latch“ genannten Registern, wobei hier der komplette Schreibzyklus, abgesehen von der Bindung der „Bitline“ an die Datenleitungen, durchlaufen wird.

### Wartezeiten

Aufgrund dieses hochkomplexen Vorgangs beim Zugriff auf Speicherzeilen und -zellen müssen zwischen den Schritten definierte Pausen eingelegt werden. Diese Wartezyklen werden als Latenzen bezeichnet und vom Speichercontroller festgelegt. Der Wert gibt die Taktzyklen an, die für bestimmte Aktionen gewartet werden muss. Da es sich um eine Zeiteinheit handelt, werden die Latenzen mit tXYZ bezeichnet, wobei XYZ die jeweils spezifische Latenz definiert.

In der Reihenfolge eines Lesezugriffs sind folgende Latenzen relevant: Zunächst bestimmt die tRP die „RAS precharge“, also die Anzahl der Taktzyklen, die vergehen müssen, bevor in einem Speicherchip eine spezielle Zeile adressiert werden kann.

Die tRAS („Row Access Strobe“) bestimmt die Anzahl der Taktzyklen,

# Wie Daten aus einem Speicherchip gelesen werden

**Im folgenden Schaubild erklären wir den grundsätzlichen Ablauf eines Lesezugriffs auf ein Speicherfeld. Die Darstellung wurde etwas vereinfacht; aktuell verwendete Speichermodule sind komplexer und bieten höhere Kapazitäten.**

1. **Adressierung**
Über die „Chip-Select"- (CS) und „Bank-Address"-Pins (BA) werden der Chip und die Bank mit der gesuchten Speicherzelle ausgewählt

2. **Zeilenadresse**
Der Speichercontroller übergibt die Zeilenadresse dieser Zelle in den Adresspuffer

3. **RAS-Signal**
Das Steuersignal RAS wird über die RAS-Pins übergeben – so erfährt die DRAM-Steuerung, dass es sich bei der Adresse um eine Zeile handelt, noch nicht um eine Spalte

4. **Zeilendekoder**
Die DRAM-Steuerung übergibt die Adresse vom Puffer an den Zeilendekoder, der nun die richtige Zeile („row") auswählt

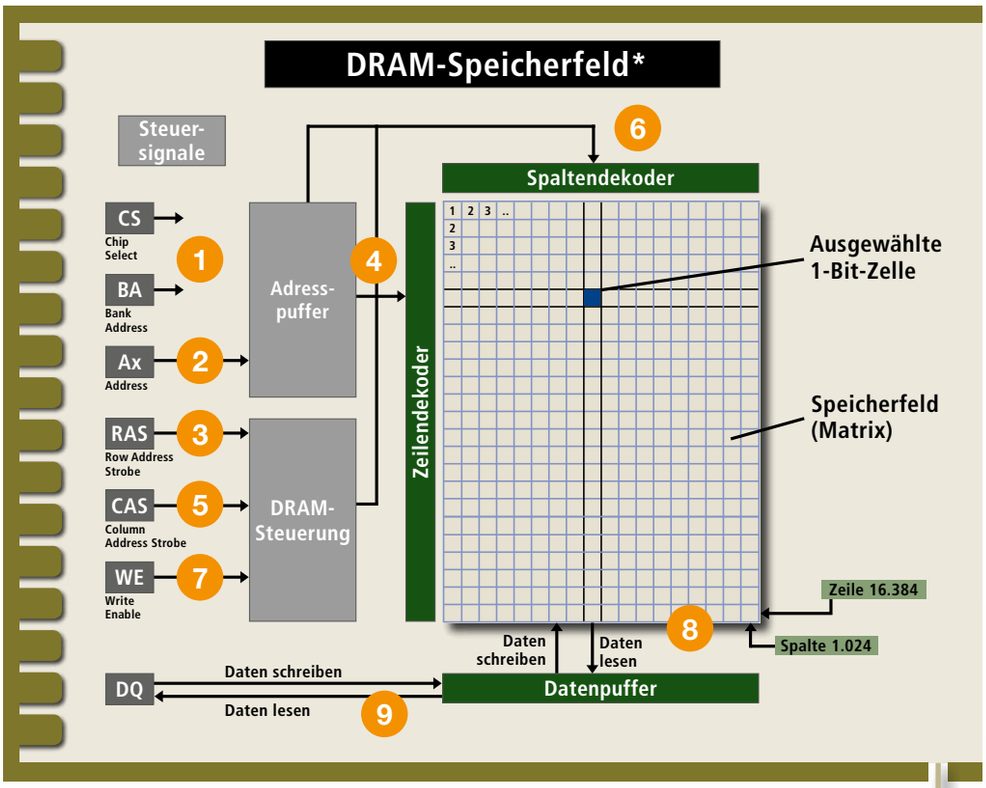
5. **Spaltenadresse (CAS)**
Nach kurzer Verzögerung folgt die Spaltenadresse mit dem CAS-Signal

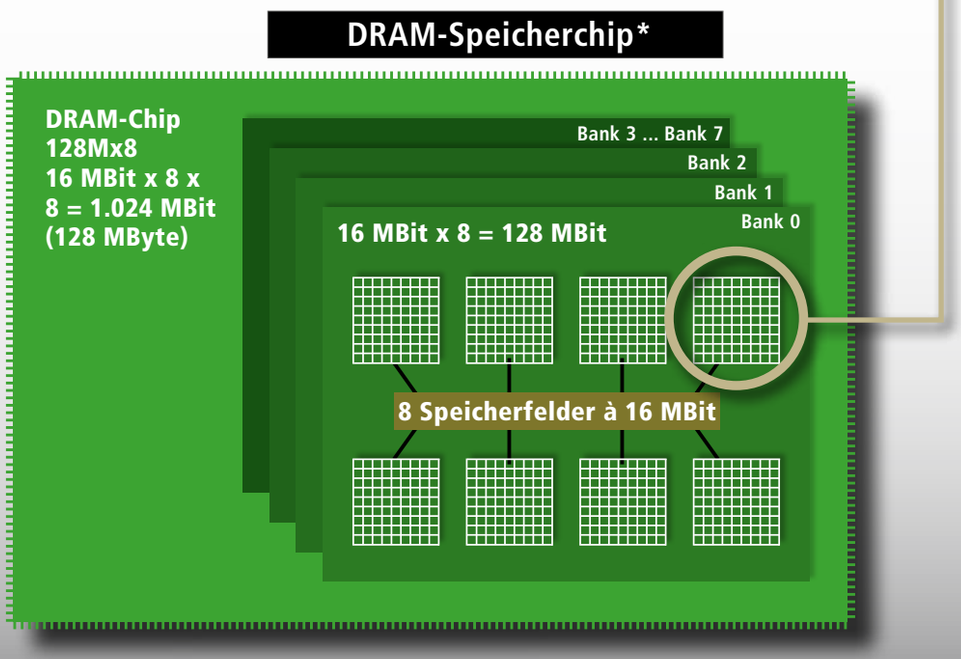
6. **Spaltendekoder**
Der Spaltendekoder wählt anhand dieser Adresse die Spalte („column") aus

7. **Lesezugriff**
Das WE-Signal („Write Enable") bleibt inaktiv. Es werden also keine Daten geschrieben, sondern nur ausgelesen; bei einem Schreibzugriff würde dieses Signal aktiviert

8. **Datenübertragung**
Die gewählte Speicherzelle entlässt ihren Inhalt über die Bitleitung in den Datenpuffer

9. **Abschluss des Zugriffs**
Der Datenpuffer übergibt die Informationen über die DQ-Pins an den Datenbus zum Speichercontroller, der sie dann an die CPU weiterreicht




* Vereinfachte Darstellung

# DRAM-Verbesserungen

Die zahlreichen Nachteile von DRAM (Zellauffrischung, vergleichsweise langsamer Zugriff etc.) sorgten schon früh dafür, dass Entwickler nach besseren Alternativen suchten. Bis heute jedoch konnte sich DDR-DRAM gegen zahlreiche Konkurrenztechniken behaupten. In jüngster Vergangenheit gab es aber einige interessante Weiterentwicklungen

## Chip-Stacking

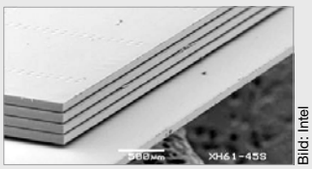


Bild: Intel

Chip-Stacking beschreibt die Anordnung mehrerer Chips übereinander, sodass auf gleicher Grundfläche zwei- bis achtmal so viel Siliziumfläche für Speicherzwecke genutzt wird. Früher wurden hierzu fertig verpackte Chips gestapelt (Package on Package), später ging man stellenweise dazu über, mehrere Silizium-Chips innerhalb eines Packages zu stapeln. Allerdings sind so nur deren Außenkanten zugänglich. Mit traditioneller Verdrahtung (wire bonding) sind so nicht genügend Kontakte für schnellen RAM möglich.

## High Bandwidth Memory

HBM wurde primär von SK Hynix in Kooperation mit AMD entwickelt und später von der JEDEC einheitlich standardisiert. Die Spezifikationen sehen bis zu acht DRAM-Speicherchips übereinander vor, zunächst erscheinen aber Varianten mit vier Ebenen. Von diesen verfügt jede über zwei DDR-Speicherkanäle, insgesamt acht, mit jeweils 128 Bit. Als Hauptspeicher-Ersatz ist HBM aber noch nicht geeignet, es kommt bislang ausschließlich auf Grafikkarten zum Einsatz.

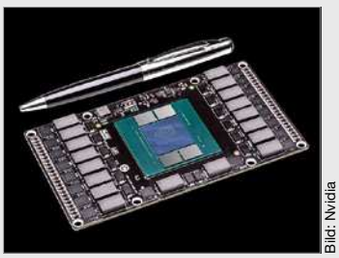


Bild: Nvidia

Die erzielte Bandbreite ist gigantisch: Vier HBM-Module erzielen 512 GByte/s Speicherbandbreite, während GDDR5-basierte GPUs maximal 385 GByte/s bei deutlich mehr Platzverbrauch schaffen. Künftige Versionen von HBM sollen die Bandbreite weiter erhöhen.

## Hybrid Memory Cube

Eine Alternative hat Micron in Kooperation mit Intel 2011 das erste Mal präsentiert. Zwischenzeitlich wurde der Entwicklerkreis zum Hybrid Memory Cube Consortium erweitert, dem unter anderem Samsung als zweiter Speicherentwickler sowie AMDs Stammfertiger Globalfoundries sowie zahlreiche Größen aus dem HPC-Geschäft wie Cray und IBM angehören.

Der Grundaufbau eines HMC ähnelt HBM. Auch die aktuellen Hybrid Memory Cubes setzen auf vier gestapelte Speicherchips, allerdings zu je 512 MiB. Als Besonderheit verlagert HMC den DRAM-Controller in den Speicherbaustein. Anstelle des massiv paralleln und komplexen DRAM-Interface, das herkömmliche DDR-DIMMs aber auch HBM-Stacks nutzen, verwendet HMC paketorientierte Übertragung und serielle Links. Verfügbar sind HMC-Varianten mit deren vier oder acht, wobei jeder Link im Normalfall aus 16 seriellen Lanes pro Richtung besteht. Für vier Links sind Geschwindigkeiten von 10 bis 15 GBit/s pro Lane bei längeren Verbindungsstrecken über das PCB spezifiziert.

Bild: Intel

die benötigt werden, um eine Zeile in der Speicherbank zu aktivieren.

tRCD („RAS low to CAS low delay“) legt fest, wie lange es dauert, bis die benötigten Spalten aktiviert sind.

Und zuletzt die bekannteste aller Latenzen, die tCL, die bestimmt, wie viele Takte vergehen müssen, bis davon ausgegangen werden kann, dass die Daten vollständig aus den benötigten Speicherzellen ausgelesen wurden.

### Stabilität contra Übertakten

Damit lässt sich auch der Zusammenhang aus Spannung, schärferen Timings und höherer Stabilität erklären. Angenommen, Sie verringern die tCL Ihres Speichermoduls. Damit nimmt der Speichercontroller bereits einen Taktzyklus vor dem vom Speicherhersteller vorgesehenen Zeitraum die Korrektheit der ausgelesenen Daten an – genauer gesagt: Der Signalpegel entspricht dann noch nicht dem Datumswert. Wird mit diesen Werten jedoch weitergerechnet, kann es zu Lese- oder Schreibfehlern und zu einem Absturz des Systems kommen.

Wenn Sie jedoch die Spannung des Speichers minimal anheben, sinkt die benötigte Zeit, um definierte Ladungszustände schneller zu erreichen. Das System kann wieder stabil werden. Das Gleiche gilt in dieser Form natürlich auch für alle anderen, hier nicht beschriebenen Latenzen.

### ECC-Speicher

Eine wichtige Funktion ist die Fehlerkorrektur bei Speichermodulen. Zu diesem Zweck kann pro Byte ein spezielles Paritätsbit zum Einsatz kommen. Ein Byte ist also fortan neun anstelle von acht Bit lang. Man spricht in diesem Fall von „ECC-Speicher“ („error correction code“).

Kippt ein Bit, was zum Beispiel durch kosmische Hintergrundstrahlung oder elektromagnetische Interferenz bei hohen Speicherdichten durchaus passieren kann, besteht mittels des Paritätsbits und des für die Korrektur zum Einsatz kommenden Hamming-Codes die Möglichkeit, diesen Fehler zu korrigieren. Kippen zwei Bits, kann dies noch erkannt, aber nicht mehr korrigiert werden, ab dem dritten kann auch ECC nichts mehr ausrichten.

ECC muss sowohl vom Speichermodul als auch vom Speichercontroller und vom Mainboard unterstützt werden. Da entsprechende Module sehr teuer und meist nur als Registered DRAM verfügbar sind, kommen sie fast ausschließlich in Server-Rechnern zum Einsatz. Alle Angaben aus den vorhergehenden Abschnitten müssen bei Registered RAMs entsprechend angepasst werden: So ist jeder „Rank“ 72-bittig angebunden und hat in der Regel neun statt acht Speicherchips.

### Datenkohärenz-Sicherung

Bei besonders hochkapazitiven RAM-Modulen kommt ein externes Register („Registered RAM“) zum Einsatz, welches zwischen dem Speichercontroller und den Speicherchips sitzt und somit den Erstgenannten „entlastet“, was die Signalqualität deutlich erhöht.

Ein Nachteil dieser Methode ist die schlechtere Leistung: Da sowohl bei Lese- als auch bei Schreibzugriffen immer zunächst in das zwischengeschaltete Register geschrieben wird, vergeht ein kompletter Taktzyklus, ehe die Daten am Ziel ankommen.

# Überblick: Von DDR1- bis DDR5-Speicher

Das grundlegende Prinzip der Datenein- und -ausgabe ist bei allen modernen Speichermodulen gleich geblieben. Wie schafften es die Entwickler, dennoch stetig neue Bandbreitenrekorde aufzustellen?

## Doppeltes Lottchen

Bei DDR-SDRAM, einer Weiterentwicklung von SDRAM, wandte man einen Trick an, um die Datenrate zu verdoppeln. Da der „Referenztakt" der Speicherchips nicht signifikant erhöht werden konnte, entschlossen sich die Ingenieure, beide Taktflanken, also die an- und absteigende, zur Übertragung von Daten zu nutzen. Dies wird durch ein Prefetch-Verfahren realisiert. So kann ein mit 100 MHz getaktetes DDR-SDRAM Modul theoretisch doppelt so viele Daten übertragen wie ein mit 100 MHz getaktetes SDR-SDRAM-Modul. In der Praxis gehen ein paar Prozent Leistung verloren.

## DDR 1 und die physikalischen Grenzen

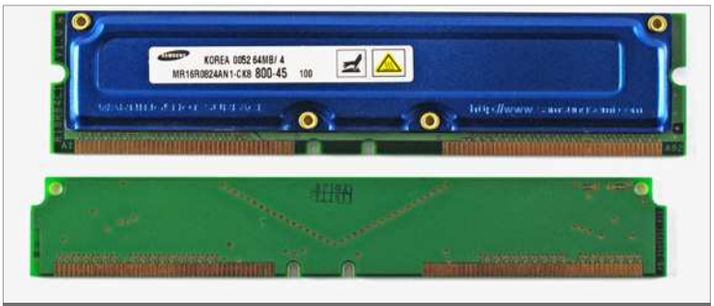
Mit den DDR400-Modulen hatten die Hersteller eine bereits mehrere Jahre alte Speichertechnologie bis aufs Letzte ausgequetscht. Einige Module benötigten zudem Kühlkörper, bis dato ein völliges Novum und zu dieser Zeit tatsächlich mehr als nur eine optische Spielerei.

Neben der enormen Wärmeentwicklung der DDR400-Module war vor allem die mangelhafte Terminierung ein großes Problem. Mit zunehmender Taktfrequenz wurde das Signal auf dem Speicherbus immer „verwaschener" (der Bus war nie für 200 MHz ausgelegt). Hintergrund: Das Signal läuft über den Speicherbus durch das RAM-Modul. Da die Terminierung – also der Abschluss des Busses – nicht direkt im letzten Chip stattfindet, sondern erst am Ende des Busses, wird das Taktsignal dort reflektiert. Das so zurückgeworfene Taktsignal „schwappt" bildlich gesprochen unkontrolliert auf dem Speicherbus auf und ab und stört so durch Übersprechung das reguläre Signal. Je höher die Taktfrequenz, desto ausgeprägter der Effekt. Hinzu kommt, dass es sich bei DDR-SDRAM um ein paralleles Arrangement handelt, das mit zahlreichen Gabelungen versehen ist. So war DDR-SDRAM im Gegensatz zu Rambus (serielle Anordnung) in Sachen maximaler Taktfrequenz von Haus aus gehandikapt.

## Der nächste Schritt: DDR2

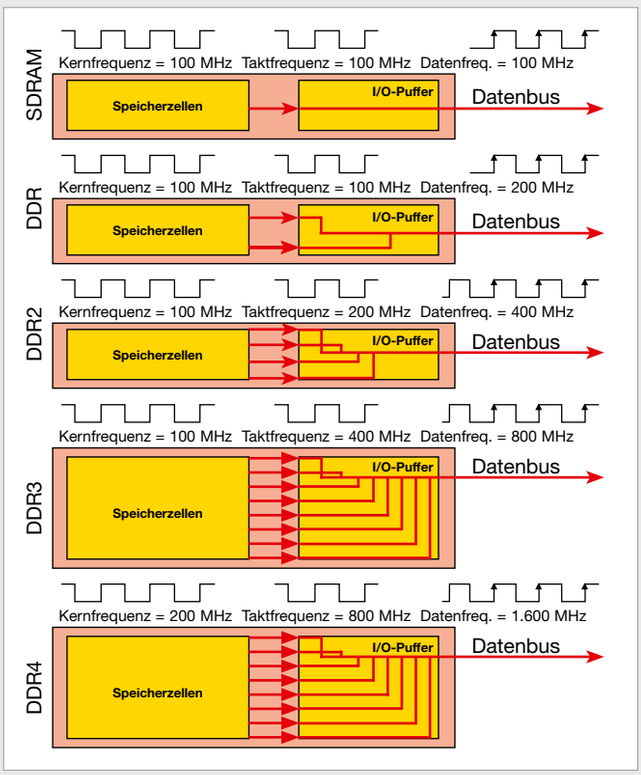
DDR2-SDRAM brachte im Vergleich zu DDR-SDRAM wieder eine Verdopplung der effektiven Datenrate. Dies wurde dadurch erreicht, dass sowohl der Prefetch-Buffer von zwei auf vier Bit als auch der I/O-Takt, mit dem das Modul nach außen kommuniziert, verdoppelt wurden. Doch wie genau funktioniert das?

Ein herkömmliches DDR400-Modul läuft real mit 200 MHz. Das gilt für den Kern des Chips ebenso wie für die Schaltungen auf dem Modul und den Speicherbus. Dank doppelter Datenrate ergibt das eine effektive Takfrequenz von 400 MHz. Bei einem DDR2-400-Modul dagegen sieht das etwas anders aus. Hier laufen die Zellen lediglich mit 100 MHz Taktfrequenz, sind jedoch mit 4 Bit Prefetches an den I/O-Puffer des Moduls angebunden statt mit nur 2 Bits. Die Zellen können also trotz halbierter realer Taktfrequenz gegenüber einem DDR-I-Modul die gleiche Transferrate liefern.



**DDR-Speicher ist nicht JEDEC-exklusiv: Schon die ersten Rambus-Module brachten Double-Data-Rate-Technik, waren allerdings deutlich teurer als vergleichbare SDR-Speichermodule jener Zeit.**

## Speichertechniken



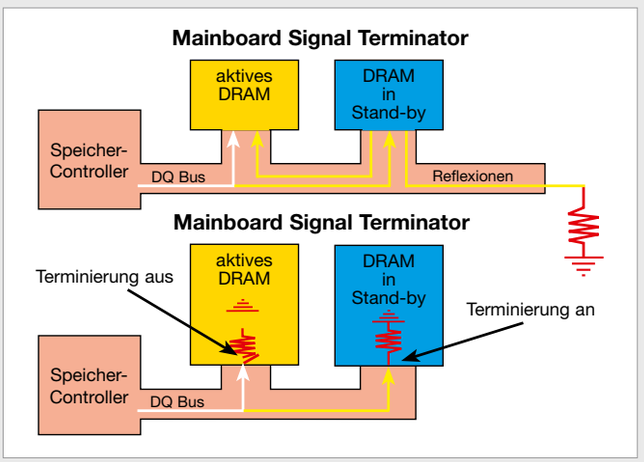

Speicherentwicklung: Lange Zeit ließ sich mehr Leistung herauskitzeln, indem die I/O-Puffer vergrößert wurden. Doch das klappte bei DDR4 nicht mehr, hier mussten die Entwickler erstmals seit fast zwei Dekaden die Kernfrequenz anheben. Die befürchtete stark erhöhte Wärmeentwicklung konnte dank feinerer Herstellungsprozesse eingedämmt werden.

# Speichertypen in der Übersicht

| Bezeichnung | Zelltakt | I/O-Takt | Datendurchsatz |
|---|---|---|---|
| **PC66** | 66 MHz | 66 MHz SDR | 533 MiByte/s |
| **PC100** | 100 MHz | 100 MHz SDR | 800 MiByte/s |
| **PC133** | 133 MHz | 133 MHz SDR | 1.066 MiByte/s |
| **DDR-200** | 100 MHz | 100 MHz DDR | 1.600 MiByte/s (PC-1600) |
| **DDR-266** | 133 MHz | 133 MHz DDR | 2.100 MiByte/s (PC-2100) |
| **DDR-333** | 166 MHz | 166 MHz DDR | 2.700 MiByte/s (PC-2700) |
| **DDR-400** | 200 MHz | 200 MHz DDR | 3.200 MiByte/s (PC-3200) |
| **DDR2-400** | 100 MHz | 200 MHz DDR | 3.200 MiByte/s (PC2-3200) |
| **DDR2-533** | 133 MHz | 266 MHz DDR | 4.266 MiByte/s (PC2-4200) |
| **DDR2-667** | 166 MHz | 333 MHz DDR | 5.333 MiByte/s (PC2-5300) |
| **DDR2-800** | 200 MHz | 400 MHz DDR | 6.400 MiByte/s (PC2-6400) |
| **DDR2-1066** | 266 MHz | 533 MHz DDR | 8.533 MiByte/s (PC2-8500) |
| **DDR3-1066** | 133 MHz | 533 MHz DDR | 8.533 MiByte/s (PC3-8500) |
| **DDR3-1333** | 166 MHz | 667 MHz DDR | 10.667 MiByte/s (PC3-10600) |
| **DDR3-1600** | 200 MHz | 800 MHz DDR | 12.800 MiByte/s (PC3-12800) |
| **DDR3-1866** | 233 MHz | 933 MHz DDR | 14.933 MiByte/s (PC3-14900) |
| **DDR3-2133** | 266 MHz | 1.066 MHz DDR | 17.066 MiByte/s (PC3-17000) |
| **DDR4-1600** | 200 MHz | 800 MHz DDR | 12.800 MiByte/s (PC4-12800) |
| **DDR4-2133** | 266 MHz | 1.066 MHz DDR | 17.066 MiByte/s (PC4-17000) |
| **DDR4-2666** | 333 MHz | 1.333 MHz DDR | 21.333 MiByte/s (PC4-21300) |
| **DDR4-3200** | 400 MHz | 1.600 MHz DDR | 25.600 MiByte/s (PC4-25600) |
| **RD300** | 300 MHz | 300 MHz QDR | 1.200 MiByte/s (PC600) |
| **RD400** | 400 MHz | 400 MHz QDR | 1.600 MiByte/s (PC800) |
| **RD3200*** | 400 MHz | 400 MHz QDR | 3.200 MiByte/s (RIMM 3200) |
| **RD4200*** | 533 MHz | 533 MHz QDR | 4.200 MiByte/s (RIMM 4200) |
| **RD6400*** | 800 MHz | 800 MHz QDR | 6.400 MiByte/s (RIMM 6400) |

*Dual-Channel

# Bus-Terminierung




Die Verlegung der Terminatoren vom Bus direkt in den Chip soll dafür sorgen, dass Signalrauschen durch Reflexionen weitgehend eliminiert werden.

Vorteil dabei: Die Zelle verbraucht wesentlich weniger Strom, da nicht nur die Taktfrequenz halbiert wird (lineare Abhängigkeit der Verlustleistung), sondern als Folge davon auch die Kernspannung von 2,5 V auf 1,8 V gesenkt werden kann (hinzu kommen allerdings meist noch bessere Fertigungsprozesse). Die Chips eines DDR2-400-Moduls produzieren demnach bis zu 70 Prozent weniger Verlustleistung als die eines vergleichbaren DDR1-400-Moduls. Oder anders gerechnet: Bei gleicher Verlustleistung können die Hersteller wesentlich höhere Taktfrequenzen aus den Chips kitzeln.

## DDR2: Schwachstellen und Lösungen

Der Preis für diese Leistungssteigerung ist ein drastischer Anstieg der Latenzen, da sich an der grundlegenden Speichertechnik der DRAM-Speicherchips nichts geändert hat. Faktisch sind die Latenzen bei DDR2-SDRAM im Vergleich zu DDR-SDRAM also fast doppelt, bei DDR3-SDRAM sogar fast viermal so hoch. Rechnet man den Takt jedoch in Nanosekunden um, ergeben sich fast identische Wartezeiten.

Auch der Signal-Reflektionen haben sich die Entwickler beim DDR2-Speicher angenommen und das Übel an der Wurzel gepackt. Getreu dem Motto „Wo nichts mehr ist, kann auch nichts mehr reflektieren“ wird der letzte Chip eines Moduls nicht mehr erst auf dem Speicherbus terminiert, sondern unmittelbar im Kern selbst. Die Entfernung zur Terminierung beträgt damit quasi null und bietet so auch kein „Futter“ mehr für Störsignale und Busrauschen.

Neben der neuen Terminierung gibt es noch eine Reihe weiterer Verbesserungen, die dafür sorgen, dass das Signal trotz hoher Taktfrequenzen sehr sauber und damit wenig anfällig für Störungen ist (zum Beispiel ein zweites Strobe-Signal und die OCD-Kalibrierung).

## DDR3-Speicher

Bei der Einführung der DDR3-Speichertechnologie hat das Speicherkonsortium JEDEC den alten Trick, der bereits bei DDR2 zum Einsatz kam, einfach wiederholt. Statt im Verhältnis 1:2 zum Speicherbus takten die Zellen auf einem DDR3-Modul nun nur noch im Verhältnis 1:4. Um trotzdem auf eine bei gleichem Speicherbus-Takt identische Transferrate zu kommen, werden bei DDR3-Speicher 8 Bits pro Prefetch ausgelesen statt 4 bei DDR2-RAM.

Allerdings war es nicht das Ziel der Entwickler, die gleiche Transferrate bei niedrigerem Takt zu erreichen, sondern eine höhere Transferrate bei gleichem Chiptakt. DDR3-1600-Module weisen dabei folgende Eckdaten auf: realer Speicherbus-Takt 800 MHz, effektiver Speicherbus-Takt dank DDR-Verfahren 1.600 MHz, Zelltakt nur 200 MHz, Transferrate ca. 12.800 MiB/s. Durch den erneut niedriger gewordenen Zelltakt bei inzwischen verbessertem Produktionsverfahren konnte die Kernspannung bei DDR3 von 1,8 auf 1,5 Volt gesenkt werden. Das kommt wiederum der Leistungsaufnahme zugute.

An einigen Schwächen ändert aber auch die DDR3-Technologie nichts: Durch den großen Prefetch-Buffer leidet die Effizienz beim Zugriff auf kleine, zusammenhanglose Speicherbereiche. Zudem gab es auch in Sachen Latenzzeit seit Jahren keine Fortschritte mehr. Ob 2T-Latenz bei DDR, 4T-Latenz bei DDR2 oder 8T-Latenz bei DDR3 – alles resultiert unter dem Strich in derselben Latenzzeit, also der Dauer in Nanosekunden, die von der Anforderung der Daten bis zur Lieferung vergehen.

## DDR4-Speicher

Der vorläufig letzte Akt beim Hauptspeicher hört auf den Namen „DDR4“. Mittlerweile setzen alle aktuellen Plattformen auf die vierte DDR-Generation. Im Wesentlichen schreibt DDR4 dabei die bisherige Entwicklung fort: Mehr Bandbreite, weniger Spannung, höher Packungsdichten, aber nach wie vor auch vergleichsweise geringe Effizienz und lange Latenzzeiten.

DDR4-Speicher arbeitet mit einer niedrigeren Standardspannung als DDR3-RAM. Erneut geht es um 0,3 Volt nach unten, und zwar auf 1,2 Volt (DDR3: 1,5 Volt). Die jüngste DDR-Generation ist darüber hinaus fit für höhere Speicherdichten und profitiert außerdem von einigen Detailoptimierungen, etwa bei der Ansteuerung und Fehlerkorrektur, damit trotz der gegenüber DDR3-RAM gesteigerten Taktraten ein stabiler Betrieb möglich ist.

## DDR4 im Detail

Die Anzahl der Kontaktflächen erhöht sich von 240 auf 288, sodass es noch wichtiger ist als bisher, sich vor dem Einsetzen zu versichern, dass der Steckplatz sowie die Kontakte frei von Verunreinigungen sind. Eine leicht auf eine Seite versetzte Einkerbung verhindert im Zusammenspiel mit einem entsprechenden Gegenstück im Steckplatz ein falsches Einsetzen.

Um mehr Speicherkapazität auf einer gleichen Chipfläche unterzubringen, spielen neben Fortschritten bei der Verkleinerung der Strukturgröße Verfahren wie 3D-Stacking eine wichtige Rolle. Mittels Silizium-Durchkontaktierung ist es möglich, eine vertikale elektrische Verbindung zwischen mehreren Lagen eines Silizium-Substrats zu realisieren und damit mehrere übereinander geschichtete Dies anzusprechen. Wie Speicherhersteller bereits gezeigt haben, ist 3D-Stacking zwar technisch bereits mit DDR3-RAM möglich, bringt aber unter anderem neue Aktivierungsbefehle und anders belegte Chip-Unterseiten mit sich, die in der ursprünglichen DDR3-Spezifikation nicht vorgesehen waren. Mit DDR4-RAM schaffte die JEDEC von Beginn an die Möglichkeit, auf 3D-Stacking zurückzugreifen und bis zu acht einzelne Silizium-Chips übereinanderzustapeln. Langfristig gesehen dürfte diese Neuerung vor allem im professionellen Umfeld geringere Aufpreise für hochkapazitative RAM-Module bedeuten.

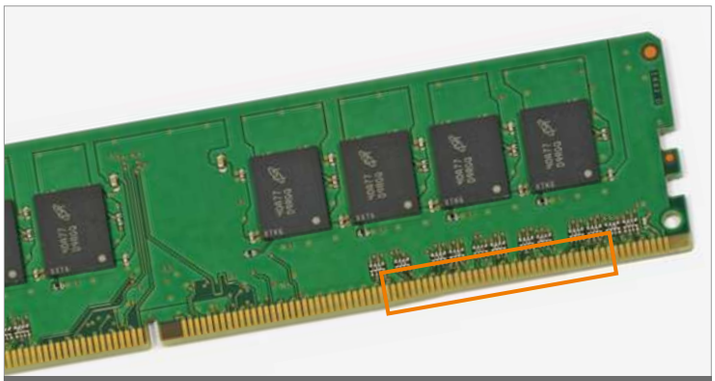
## Doppelte Dichte

Mit DDR4 erhöht sich auch die Komplexität der internen Organistation weiter: Bis zu 8 Ranks und 16 interne Bänke – nicht mit Steckplätzen auf dem Mainboard zu verwechseln – sind nun erlaubt; bei DDR3-RAM lagen die Limits noch bei 4 Ranks und 8 internen Bänken. Die 16 Bänke eines Speicherchips werden mit DDR4 allerdings in Gruppen von 4 Bänken unterteilt.

Jede Bank ist unterteilt in Zeilen und Spalten. Die Speicherkapazität pro Zeile beträgt bei DDR3-RAM entweder 1 oder 2 KiByte, bei DDR4-RAM wurde die Untergrenze auf 512 Byte gesenkt. Je kürzer eine Zeile, desto schneller können unterschiedliche Zeilen aktiviert werden und desto weniger Energie wird benötigt.

## Detailverbesserungen

Einmal mehr macht die Signalqualität bei der nächsten Ausbaustufe Probleme. Daher beinhaltet die DDR4-Spezifikation mehrere Maßnahmen, welche einen fehlerfreien Betrieb sicherstellen sollen. DDR4-RAM unterstützt beispielsweise eine zyklische Redundanzprüfung bei Schreibzugriffen sowie die Paritätsprüfung von Kommando- und Adressierungssignalen.



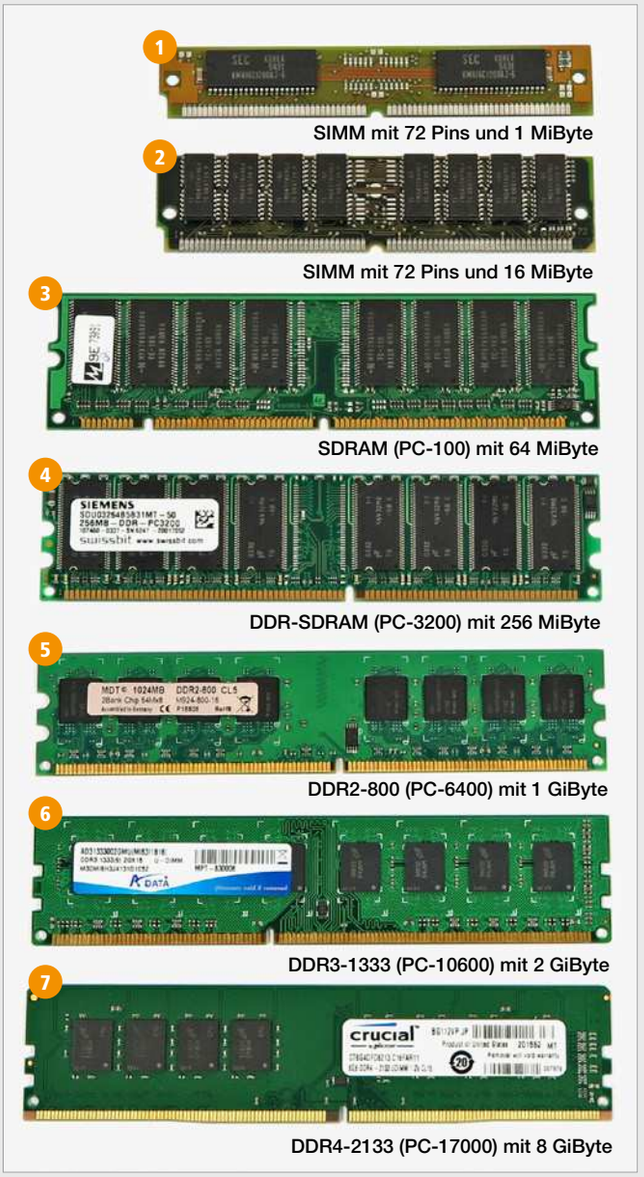
Alle DDR4-Module zeichnen sich durch eine zur Seite hin abgeschrägte Kontaktleiste aus – so auch dieser mit Micron-D9RGQ-Chips bestückte 8-GiByte-Riegel von Crucial.

## DDR4- und DDR3-RAM im Vergleich

| Eigenschaft | DDR3-RAM | DDR4-RAM |
|---|---|---|
| **DDR-Rating** | DDR3-800 bis DDR3-2133 | DDR4-1600 bis DDR4-3200 |
| **Zelltakt** | 100 bis 266 MHz | 200 bis 400 MHz |
| **E/A-Puffer-Takt** | 400 bis 1.066 MHz | 800 bis 1.600 MHz |
| **Kapazität pro Chip** | 512 Mbit bis 8 Gbit | 2 Gbit bis 16 Gbit |
| **Kapazität pro Modul** | max. 128 GiByte (theoretisch) | max. 512 GiByte (theoretisch) |
| **Bänke pro Speicherchip** | 8 | 16 (je 4 gruppiert) |
| **Kontakte** | 240 | 288 |
| **Modulhöhe** | 30,35 Millimeter | 31,25 Millimeter |
| **Modulbreite** | 133,35 Millimeter | 133,35 Millimeter |
| **Moduldicke** | 1,00 Millimeter | 1,20 Millimeter |
| **Standardspannung** | 1,50 Volt | 1,20 Volt |
| **Low-Voltage-Option** | 1,35 Volt (DDR3L) bzw. 1,25 Volt (DDR3U) | 1,05 Volt (DDR4L), aber noch nicht spezifiziert |
| **Prefetch** | 8 Bit | 8 Bit |

# Überblick: Speichermodule im Vergleich

Eine Auswahl aus rund 30 Jahren Speichergeschichte von alt nach neu: SIMM mit 72 Pins und 1 MiByte **(1)** sowie 16 MiByte **(2)** mit jeweils 60 ns Zugriffszeit, SDRAM (PC-100) mit 64 MiByte **(3)**, 256 MiByte DDR-SDRAM (PC-3200) **(4)**, 1 GiByte DDR2-800 (PC-6400) **(5)**, 2 GiByte DDR3-1333-RAM (PC-10600) **(6)** und 8 GiByte DDR4-2133 (PC-17000) **(7)**




Neu ist auch die im Speichersegment bereits mit GDDR4 eingeführte Technik „Data Bus Inversion“ (DBI). Dabei werden eingehende Signale elektrisch invertiert, also umgedreht, wenn sich damit ein niedrigerer Energiebedarf und weniger Ladungswechsel erzielen lassen. Eine 0 wird also zu einer 1, wenn bei einer eingegangenen Gruppe von Bits mindestens die Hälfte eine Ladungsänderung zur korrekten Verarbeitung erfordern würde. Die betroffene Folge von Bits wird allerdings mithilfe eines Flag-Bits gekennzeichnet, damit die Daten später korrekt ausgelesen, also erneut invertiert werden. Dieser Vorgang verbessert in der Praxis die Signalqualität und reduziert die Leistungsaufnahme.

## Referenzspannungen

Referenzspannungen stellen vereinfacht gesagt die Grenze zwischen den Spannungszuständen dar, die entweder als 0 oder als 1 interpretiert werden können. Bei DDR3 gibt es zwei verschiedene Eingangs-Pins für den Daten-Bus sowie für Kommando- und Adressierungssignale. Bei Datenübertragungen entsteht allerdings Signalrauschen, das auch die Referenzspannungen beeinträchtigt. Daher sieht die JEDEC für DDR4-RAM vor, dass der Eingangs-Pin für die Referenzspannung der Datenleitung wegfällt und diese Referenzspannung stattdessen intern erzeugt wird.

Bei der Eingabe-/Ausgabe-Schnittstelle der Datenleitung kommt zur Signalterminierung statt SSTL (Series-stub terminated logic) nun POD-Technik (Pseudo-open drain) zum Einsatz, die im Grafikbereich bereits seit GDDR3 üblich ist. Abhängig von der Temperatur finden zum Erhalt des Speicherinhalts erforderliche Wiederauffrischungszyklen nun seltener statt – je kühler, desto seltener.

## Kein neues Prefetching

Positiv dürfte sich auf die Leistung auswirken, dass anders als bisher bei jeder Einführung eines neuen DDR-RAM-Standards nicht noch mehr Daten auf Verdacht in einen Puffer geladen werden (Prefetching) und der Eingabe-/Ausgabe-Puffer beschleunigt wird – dessen Takt gibt PC Games Hardware üblicherweise an –, sondern stattdessen die Taktfrequenz der Speicherzellen selbst gesteigert wird.

Ein Beispiel: Während die Speicherzellen eines DDR2-800-Riegels und eines DDR3-1600-Moduls jeweils mit 200 MHz laufen, beträgt der Zelltakt eines DDR4-3200-Sticks 400 MHz. Eine erneute Ausweitung des Prefetching hätte dazu geführt, dass sich die Effizienz von DDR-RAM weiter reduziert, denn schließlich werden nie alle auf Verdacht vorgeladenen Daten tatsächlich benötigt.

## DDR4 und die Zukunft

Den großen Schnitt macht DDR4-Speicher nicht; allerdings bietet die neue Generation einige Detailverbesserungen, die für mehr Speicherkapazität, höhere Geschwindigkeit und weniger Energiebedarf sorgen. Ob und wann es einen komplett neuen Speichertyp geben wird, der viele Unzulänglichkeiten von DDR-RAM beseitigt, weiß derzeit niemand.

Als wahrscheinlich gilt es, dass die parallele Datenübertragung von und zum Speicher wie bei vielen anderen Schnittstellen mittelfristig einer seriellen wird weichen müssen. Die Begriffe seriell und parallel wecken dabei allerdings falsche Assoziationen. Gemeint ist lediglich, dass die Speichereinheiten so organisiert werden könnten, dass simultane Zugriffe einfacher werden. Eine solche Technik dürfte allerdings nicht als „DDR5“ auf den Markt kommen.

# Infrastruktur

# So funktioniert die Infrastruktur: Solid State Disks

Einige SSDs (hier eine Sandisk Ultra Plus) kommen mit vier Speicherchips aus; der zur Verfügung stehende Raum wird nicht einmal ansatzweise ausgenutzt, was letztlich neue und kompaktere Bauformen wie M.2 ermöglichte (siehe auch Kapitel „Mainboards“)

SSDs, also „Solid State Drives“, entwickeln sich rasant weiter: Innerhalb weniger Jahre haben sie sich von Laufwerken, für die SATA 1,5 GBit/s noch ausreichend war, zu Speichermedien gemausert, die lange Zeit jeden Schnittstellenstandard vor sich hertrieben (siehe auch Kapitel Mainboard, Abschnitt SATA).

## Das SSD-Prinzip

Die grundlegende Arbeitsweise ist bei fast allen SSDs gleich. Hinter der Schnittstelle zum System wartet der Controller, der die Verteilung der Daten regelt. Die Daten selbst lagern auf den Flash- oder auch NAND-Chips.

Im Gegensatz zu normalem Speicher behalten sie ihren Dateninhalt auch, wenn gerade kein Strom anliegt – „nichtflüchtig“ oder englisch „non-volatile“ nennt man das in der Fachsprache.

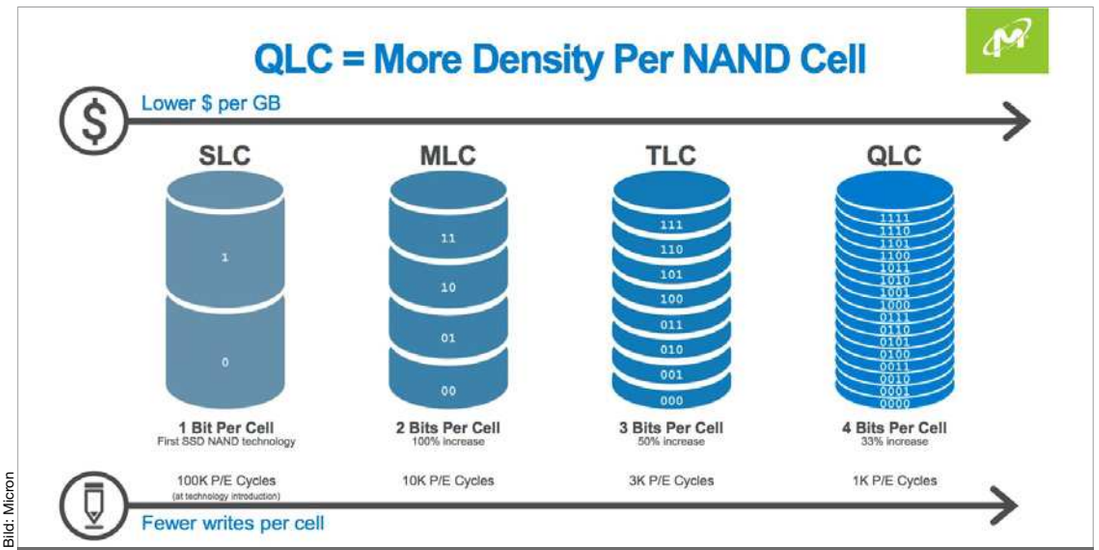



Bild: Micron

Die Speicherdichte erhöht sich mit jedem zusätzlichen Bit pro Zelle. Die 16 Spannungsniveaus pro Zelle eines QLC-Flashs könnten aber die letzte Evolutionsstufe des NAND-Speichers sein

## Controller und Cache

Der Controller entscheidet darüber, wie die Bits bestmöglich auf die einzelnen Flash-Speicher verteilt werden. Beim Controller-Chip handelt es sich um einen per Firmware gesteuerten Prozessor. Oft basiert dieser auf dem CPU-Design der britischen ARM-Holding. Die meisten modernen Controller nutzen einen schnellen Cache, der aus denselben Speicherbausteinen wie normaler Arbeitsspeicher besteht – von DDR- bis DDR4-RAM ist alles vertreten.

## Sandforce-Controller

Eine Ausnahme bilden die Sandforce-Controller. Diese weisen einige Eigenschaften auf, die sie eine zeitlang zum beliebtesten SSD-Controller überhaupt machten, inzwischen aber zu einem Nachteil statt einem Vorteil wurden. Die Sandforce-Familie wurde ursprünglich nämlich für günstige SSDs entwickelt. Sie ermöglicht den Verzicht auf den DDR-Cache. Die fehlende Caching-Möglichkeit gleichen die Sandforce-Controller durch eine integrierte Kompressionsfunktion aus, die eintreffende Daten vor dem Schreiben komprimiert. Da sich so die zu sichernde Datenmenge verringert, schien es zu Zeiten von langsamerem Flash so, als wären Sandforce-SSDs besonders leistungsfähig.

Nun, einige Jahre später, wo alle Controller die von SATA-6-GBit/s herangekarrten Datenmengen mehr oder weniger mühelos in Echtzeit aufs Flash bannen können, hat sich der Vorteil zu einem Nachteil verkehrt. Die Kompressionsfunktion ist nämlich nutzlos, wenn die ankommende Datenmenge gedeckelt ist, und der Controller und das Flash die ankommenden Daten gut abarbeiten können. Zudem lässt sich die Kompression nur auf unkomprimierte Inhalte anwenden.

Damit fallen viele Mediendateien durch das Raster – ausgerechnet jener Bereich, wo das höchste Datenaufkommen auftritt.

## Controller-Eigenheiten

Daten können bei SSDs wie Festplatten immer nur blockweise geschrieben werden. Auch wenn nur ein einzelnes Bit geändert werden muss, wird eine „Page“ in den Controller-Cache eingelesen, dort umprogrammiert und dann am Stück zurückgeschrieben. Daher ist es enorm wichtig, dass der Controller die Zugriffe sinnvoll und möglichst optimal zusammenfassen kann.

## Kanäle und Chips

Darüber hinaus werden mehrere Datenleitungen vom Controller parallel verwaltet und bedient. Eine solche Datenleitung wird auch als Kanal bezeichnet; aktueller Standard sind vier bis zehn Kanäle pro SSD-Controller. Die meisten Controller verwenden acht Kanäle, technisch möglich sind bei der aktuellen Generation aber 4 bis 32.

Pro Kanal kann wiederum eine bestimmte Menge Chips – meist ebenfalls vier bis acht an der Zahl – angebunden werden. Die Anzahl der Chips kann mit entsprechenden Dekodern auf bis zu 16 pro Kanal erhöht werden (siehe auch Darstellung in der Randspalte).

## Volle Leistung

Um die volle Leistung zu erreichen, muss beim Controller eine Mindestzahl an Kanälen mit Flash-Chips bestückt sein. Die Kapazität ist dabei unwichtig, alleine die Anzahl der bestückten Kanäle sowie die Geschwindigkeit der Chips ist relevant.

Wenn wir pro Flash-Package eine maximal unterstützte Geschwindigkeit von 1.600 MBit/s (ein Byte entspricht 8 Bit, also 200 MByte/s) ansetzen, brauchen wir theoretisch insgesamt vier Chips an vier Kanälen, um ein SATA-6-Gbit/s-Interface voll auszulasten. Dieses bietet maximal 750 MByte/s abzüglich Overhead, den wir der Einfachheit halber aber ignorieren. Sind die Flash-Chips langsamer, muss zusätzlicher Speicher angebunden werden, um in Summe die gleiche Leistung zu erreichen.

SATA-6 Gbit/s stellt dabei aber erst seit Kurzem den Flaschenhals dar. In den Jahren zuvor war es meist der SSD-Controller.

## Flash-Speichertypen

Neben der Organisation kommt es natürlich auch auf die Art und Eigenschaften des Speichers selbst an. Eines der wichtigsten Kriterien für die Performance ist die Zahl der gespeicherten Bits pro Speicherzelle. Unterschieden wird dabei nach SLC, MLC, TLC- und QLC-Zellen. SLC-Flash-Bausteine können pro Zelle genau ein Bit speichern („Single Level Cell“), MLCs dagegen mindestens zwei Bits pro Zelle („Multi Level Cell“). Seit einiger Zeit verwenden die Hersteller auch TLC-Speicher, der drei Bits aufnehmen und so insgesamt acht verschiedene Zustände unterscheiden kann („Triple Level Cell“). Zur besseren Abgrenzung benutzt man den Begriff „MLC“ seitdem synonym für Zwei-Bit-Speicherzellen. Auch die nächste Ausbaustufe QLC („Quadruple Level Cell“), ist bereits im Einsatz.

Alle Zelltypen haben Vor- und Nachteile. SLC-Speicher ist am schnellsten, erlaubt aber keine allzu großen Packungsdichten. MLC, TLC und QLC dagegen vertragen aufgrund der feineren Bauweise weniger Schreibzyklen. Die Hersteller unternehmen daher Schritte, um die Lebensdauer der Zellen unter Beibehaltung der Speicherdichte zu erhöhen. Erhöhte Ausfallraten mit MLC- und TLC-Flash sind immerhin bislang nicht bekannt geworden.

## 3D-VNAND

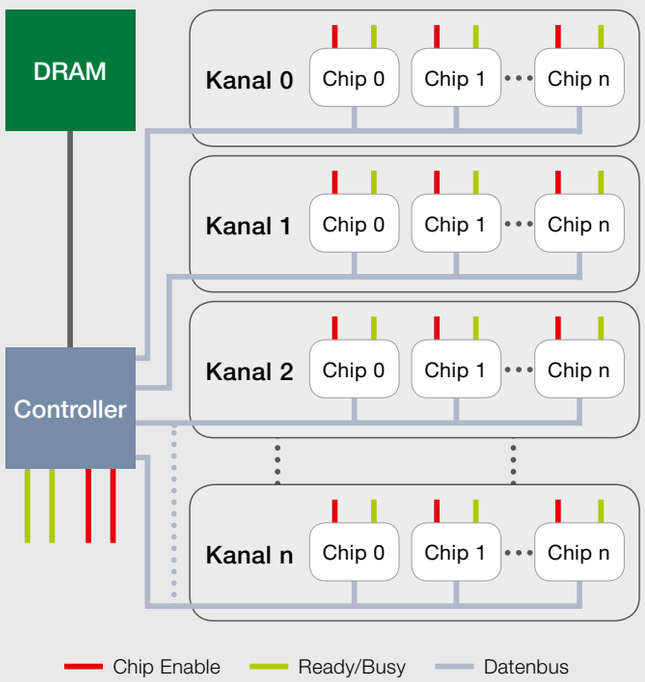
Der Ausweg aus dem Dilemma, für immer höhere Speicherdichte zu sorgen und gleichzeitig die Strukturen widerstandsfähig genug für ein ganzes SSD-Leben zu halten, ist derselbe, wie ihn Architekten für die Gebäude in den Großstädten dieser Welt gefunden haben: Statt in die Breite zu wachsen, wird der Chip in die Höhe gebaut. Samsung nennt es 3D-VNAND („Vertical NAND“) und wandte es erstmals bei Datacenter-SSDs und der SSD 850 Pro an. Das Unternehmen stapelt 32 Lagen des Speichers übereinander, sodass die Strukturen in den einzelnen

## Wie ein modernes SSD-Laufwerk aufgebaut ist

**Ein Flash-Chip ist bei Weitem nicht so schnell, wie es die Leistungsdaten aktueller SSDs vermuten lassen. Stattdessen resultiert die Geschwindigkeit aus einer intelligenten Verschaltung von vielen Chips.**

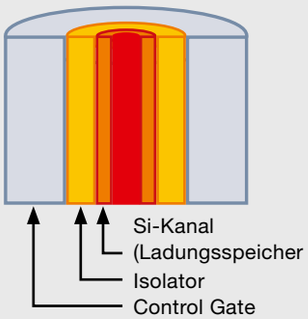
Ausgehend von dem Maximaldurchsatz eines aktuellen Controllers besitzt ein Flash-Chip eine Geschwindigkeit von bis zu 200 MB/s. Um eine höhere Geschwindigkeit zu erreichen, werden mehrere davon verschaltet. Dabei lässt sich eine gewisse Anzahl Chips pro Kanal betreiben.

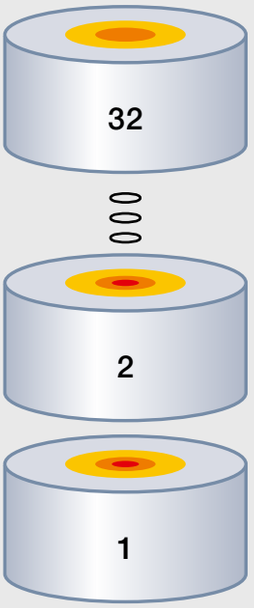
Die Flash-Chips sind über den Datenbus mit dem Controller verbunden. Über diesen senden und empfangen sie die Daten, welche gespeichert werden sollen oder die der Controller angefordert hat. Damit die Chips, welche am gleichen Kanal hängen, nicht gleichzeitig senden oder empfangen, stehen sie zusätzlich über Steuerleitungen mit dem Controller in Verbindung. Über diese kann der Controller die einzelnen Chips (de-)aktivieren und der Chip kann seinen Status dem Controller mitteilen. So kann dieser Schreib- und Lesevorgänge abwarten und bei Bedarf hereinkommende Daten auf einen anderen Kanal umleiten.



## 3D-VNAND erklärt

**Da der Bedarf an Speicherdichte immer weiter zunimmt, der Integrationsprozess dafür aber nicht einfacher wird, dürfte sich zukünftig das Konzept des 3D-VNAND durchsetzen. Durch die runde Form der Speicherzelle lassen sich viele davon übereinanderstapeln – zurzeit bis zu 96 davon. Da die Ladung von der röhrenförmigen Control-Gate-Siliziumschicht eingeschlossen ist, ist sie besser vor Interferenzen geschützt.**




3D V-NAND

Schichten wieder wachsen dürfen. Aber das ist nicht der einzige Vorteil: Während die Speicherzellen bislang meist in einer flachen Bauform ausgeführt sind, haben sie im VNAND eine runde Form. Dadurch sind die Speicherzellen nach außen abgeschirmt, was mögliche Interferenzen reduziert. Die zur Fehlerkorrektur notwendige Logik darf einfacher ausfallen und arbeitet schneller, was auch in einen Vorteil für die SSD-Leistung mündet.

### „Pseudo-SLC“

MLC-Speicher benötigt für das Beschreiben von Zellen beziehungsweise das Speichern von Daten deutlich mehr Zeit als SLC-basierter Speicher. Um diesen Nachteil auszugleichen, beschreiben einige Laufwerke bestimmte Bereiche einer MLC-SSD wie SLC-Speicher mit nur einem Bit. Es handelt sich also um eine Art „Pseudo-SLC“, das wie eine zweite Cache-Stufe funktioniert.

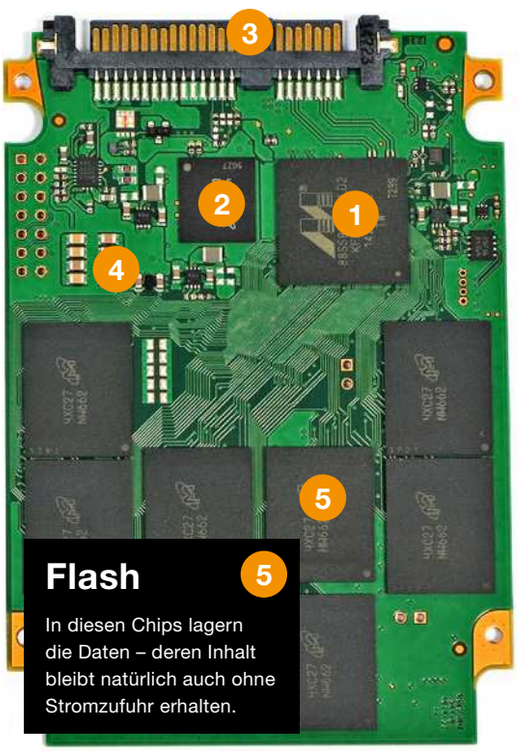
Während es bei Sandisk und Samsung festgelegte Bereiche mit fixer Kapazität sind, arbeitet eine relativ neue Lösung von Crucial dynamisch. Konkret bedeutet das, dass jede Speicherzelle in der SSD theoretisch als Pseudo-SLC dienen kann. Zudem darf diese Funktion (vorübergehend) bis zur Hälfte des SSD-Speicherplatzes verbrauchen. Etwaige arbeitsfreie Momente nutzt der Controller dann, um später den eigentlichen Wunschzustand in Form von beschriebenen MLC-Zellen herzustellen.

### Wear Levelling & Co.

Da die einzelnen Flash-Zellen nur eine begrenzte Anzahl an Schreibzugriffen überstehen, werden Vorkehrungen getroffen, damit die Belastung durch Schreibzugriffe den Flashspeicher möglichst gleichmäßig „abnutzt“ (Wear Levelling). Dazu gehört neben dem Umsortieren häufig verwendeter Daten zum Beispiel auch das Zusammenfassen von Zugriffen.

Daneben wird oft noch ein Bereich des Flashspeichers in Reserve gehalten, in dem freie Blocks zur Verfügung stehen, wohin die Zugriffe umgeleitet werden können – sowohl das Wear Levelling als auch die Performance kann hiervon profitieren. Schließlich steht mit dem Trim-Befehl unter aktuellen Betriebssystemen eine Möglichkeit zur Verfügung, der SSD zu signalisieren, welche Blocks nicht mehr benötigte Daten enthalten und direkt überschrieben werden können, ohne sie vorher einlesen und separat löschen zu müssen. Allzu große Sorgen um die Flash-Lebensdauer muss sich aber niemand machen: Selbst billige SSDs überstehen bei normalem Gebrauch locker zehn Jahre und mehr.

## Innenleben einer SSD (SATA)




**Controller** 1

Hierbei handelt es sich um das Herzstück der SSD, maßgeblich verantwortlich für die Performance.

**Cache** 2

Je nach Controller-Typ wird DRAM-Cache genutzt, um die Zugriffe auf den Flash-Speicher zu beschleunigen.

**Schnittstelle** 3

SATA 6 Gb/s war lange Zeit die Standard-Schnittstelle; der Wachwechsel (zu PCIe) ist aber in vollem Gange.

**Puffer** 4

Stützkondensatoren verhindern bei besseren Modellen einen augenblicklichen Datenverlust bei Stromschwankungen

**Flash** 5

In diesen Chips lagern die Daten – deren Inhalt bleibt natürlich auch ohne Stromzufuhr erhalten.

# So funktioniert die Infrastruktur: Festplatten

Festplatten sind der Inbegriff der sogenannten Massenspeicher. Diese sind trotz der Konkurrenz durch SSDs immer noch am preiswertesten (pro Gigabyte) – wenngleich sie in nahezu allen anderen Disziplinen das Nachsehen haben.

## Speicherriesen

Festplatten werden nach wie vor recht schnell weiterentwickelt. So sprang die Kapazitätsgrenze in einem 3,5-Zoll-Laufwerk von vier Terabyte im Jahre 2014 auf acht Terabyte Ende 2015. 2017 waren bereits 12 TB erreicht und seit 2019 sind Modelle mit 14 Terabyte im Endverbraucher-Segment erhältlich. Neue Schreibtechnologien sollen die Speicherdichte von Festplatten weiter erhöhen. Mit dem kürzlich vorgestellten „Heat-assisted magnetic recording“ (HAMR) sollen in Zukunft HDDs möglich werden, die eine Kapazität von 100 Terabyte und mehr aufweisen können.

## Das Grundprinzip

Vor mehr als 60 Jahren hat alles begonnen: Die IBM 350 RAMAC war die erste Festplatte, sie konnte ganze 3,75 Megabyte Speicher bereitstellen. Dafür benötigte man 1956 einen 173 cm hohen Plattenstapel aus 50 magnetisierbaren Aluminiumscheiben.

Solche Plattenspeicher sind auch mehr als 60 Jahre später noch im Einsatz, natürlich viel geringer dimensioniert und deutlich leistungsfähiger als damals, das Grundprinzip hat sich aber seither nicht wesentlich verändert. Durch neue Materialien und Herangehensweisen beim Schreiben konnte die Speicherdichte immer wieder erhöht werden.

## Der Grundaufbau

Prinzipiell werden Daten auf allen Festplatten durch magnetische Zustände gesichert. Gespeichert wird auf magnetischen Metallscheiben, die mit Hochgeschwindigkeit um eine Achse rotieren. Diese Scheiben bestehen meist aus einer Aluminium- oder Magnesium-Legierung, aus Glas oder Verbundstoffen – je nachdem welche Datendichten man erreichen möchte. Wichtig ist, dass die Materialien formstabil sind, eine geringe elektrische Leitfähigkeit aufweisen und nicht magnetisch sind.

Diese Metallplatten (engl. Platter) werden dann mit einer Eisenoxid- oder Kobalt-Schicht versehen, welche die eigentliche Speicherung von Informationen erlaubt. In dieser Schicht sind winzige Partikel vorhanden, die in Körnern eine magnetische Polung annehmen und so Daten speichern können. Die beiden unterschiedlichen magnetischen Ausrichtungen entsprechen dabei den Einsen und Nullen binärer Daten.

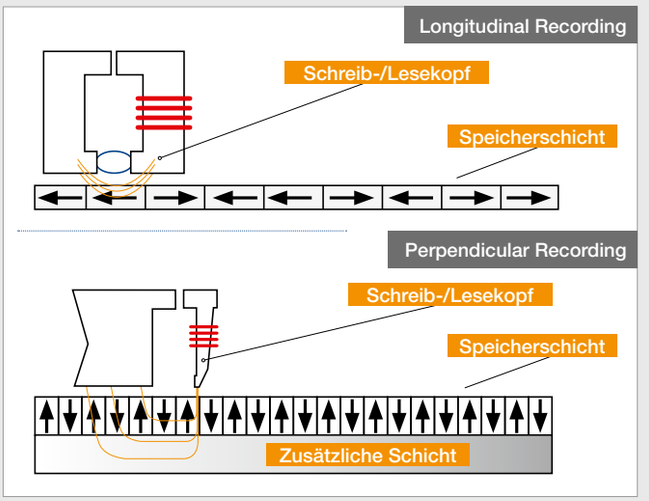
Prinzipiell könnte eine Speicherung von Information bereits jetzt erfolgen, aber in der Praxis wird dieser magnetisierbare Belag noch zusätzlich mit einem diamantähnlichen Kohlenstoff geschützt, um mechanische Beschädigungen durch den Lese- und Schreibkopf des Massenspeichers zu vermeiden. Außerdem erhöht dies die Kapazität der HDD weiter.

Bei hohen Speicherdichten und Umdrehungsgeschwindigkeiten von mehreren tausend Rotationen pro Minute ist die Oberfläche und allgemein die innere Mechanik einer Festplatte extrem empfindlich.

Formfaktoren: 5,25 Zoll, 2 x 3,5 Zoll (PATA/SATA), 2,5 Zoll (für Notebooks), 1,8 Zoll (Ultra-Portables/Netbooks) und ein 1-Zoll-Microdrive in aufsteigender Reihenfolge

## Wie die Festplattenkapazität mit PMR erhöht wird

**Perpendicular Recording ist eine Technik, mit der die Datendichte erhöht werden kann. Unsere Übersicht zeigt, wie es funktioniert**




**Zwar ist der technische Aufwand beim „Perpendicular Recording“ höher, die vertikale Ausrichtung der Speicherpartikel erlaubt dafür eine deutlich höhere Datendichte**

Festplatten, die mit Helium gefüllt sind, wirken wie aus einem Guss. Charakteristisch für sie sind die fehlenden Schrauben an der Oberseite des Gehäuses

Daher gleicht das Innenleben der Festplatte quasi einem Reinraum. Kleinste Mengen Staub zwischen Platter und Lesekopf oder gar das Öffnen des Gehäuses sind in der Regel tödlich für HDDs.

### Helium

Mittlerweile kann man Festplatten auch gar nicht mehr öffnen: Der Hersteller HGST (Hitachi) brachte vor einigen Jahren mit Helium gefüllte Festplatten auf den Markt. Das Edelgas hat eine geringere Dichte als Luft, was es möglich macht, Platten enger übereinander zu setzen, da dadurch Strömungseffekte minimiert werden. So können mehr Scheiben verwendet werden, was die Kapazität erhöht. Zudem sollen die Festplatten dadurch auch etwas kühler laufen, was letztlich auch Strom spart.

Laut HGST sollen dem mehrere Jahre Entwicklungszeit vorausgegangen sein, da es nicht nur eine Herausforderung war, das Gas in das Gehäuse hineinzubekommen, sondern es auch so zu verschließen, dass es nicht wieder entweicht. Mittlerweile sind Festplatten mit hoher Kapazität aller Marken mit Helium gefüllt.

### Festplatten-Geometrie

Doch auch Im Helium-Reinraum entfällt das Grundprinzip der Festplatte nicht. Die Daten sind und bleiben, ähnlich wie auf einer Schallplatte, in konzentrischen Ringen, nicht einer fortlaufenden Rille, auf den Plattern angeordnet. Zwar ist ein separater Kopf pro abzutastender Platteroberfläche vorhanden, bewegt werden kann aber nur das gesamte Konstrukt auf einmal – und dies kostet einfach vergleichsweise viel Zeit und Energie.

Da überrascht es nicht, dass die Entwickler schon relativ bald auf die Idee kamen, Schreibvorgänge umzusortieren und damit die Performance zu erhöhen. Mit NCQ („Native Command Queue“) werden unnötige Bewegungen des Schreib-Lese-Kopfes vermieden und die Wartezeiten verkürzt, bis der Kopf auf der richtigen Position der Scheiben steht. Für NCQ ist das AHCI-Protokoll erforderlich.

### LMR und PMR

Die Kapazität einer Festplatte wird logischerweise durch die Fläche begrenzt, die magnetisiert werden kann. Das Longitudinal Magnetic Recording (LMR, längslaufende magnetische Aufzeichnung) wurde nahezu 50 Jahren für die Speicherung von Informationen auf einer Festplatte verwendet und erst 2006 durch den Nachfolger, die Senkrechtaufzeichnung (PMR), abgelöst.

Beim LMR sind die Bereiche, die für ein Bit an Information magnetisiert werden, parallel zur Rotationsrichtung des Datenträgers orientiert. Das Schreiben von Daten erfolgt dann durch einen Schreibkopf, der über eine Spule verfügt, die mit dem entstehenden Magnetfeld den Bereich direkt darunter auf der Metallplatte magnetisiert.

### Schreibvorgang

Beim Schreibvorgang entstehen so viele kleine magnetisierte Bereiche, die Daten konzentrisch auf der Scheibe anordnen. Dabei entstehen auch benachbarte Bits mit einer entgegengesetzten Magnetisierung, die durch eine Übergangsschicht voneinander getrennt vorliegen. Beim Lesevorgang erzeugen die kleinen magnetischen Bereiche auf der Festplatte ein Magnetfeld in der Spule, sodass es zur Induktion einer geringen Spannung kommt, die verstärkt wird und dann als Datenstrom ausgelesen werden kann.

Durch Verringerung der Speicherstruktur – also kleinere magnetisier-

bare Bereiche, die ein Bit darstellen – können so Datendichten von bis zu 31 Gigabit pro Quadratzentimeter erreicht werden. Erst dann wirkt sich der superparamagnetische Effekt aus.

Der Superparamagnetismus stellt für magnetisierbare Speichermedien die physikalische Obergrenze der möglichen Aufzeichnungsdichte dar, da sich hier schon geringe Temperaturschwankungen auf die Daten auswirken. Die Speicherkörner an magnetisierbaren Partikeln können die Magnetisierung bereits durch geringe thermische Änderungen verlieren.

## Perpendicular Magnetic Recording

Um die Kapazität der bisherigen Festplatten weiter zu erhöhen, kam man auf die pfiffige Idee, die magnetisierbaren Bereiche nicht längs, sondern senkrecht zu organisieren. Mit der neuen Anordnung der Speicherkörner beim sogenannten Perpendicular Magnetic Recording (PMR) konnte die Datendichte bei gleicher Fläche etwa dreimal so dicht organisiert werden wie noch beim LMR. Damit sind bis zu 160 Gigabit pro Quadratzentimeter realisierbar.

Bereits 1976 stellte Shun'ichi Iwasaki fest, dass die senkrechte Ausrichtung des magnetischen Moments bei Datenträgern eine Erhöhung der Speicherdichte nach sich zieht, aber erst 2005 war es möglich, diese Technik als Konsumgut in Form einer 2,5-Zoll-Festplatte mit 160 GB und einer 3,5-Zoll-Festplatte mit 750 GB Speicher unterzubringen – ganze 29 Jahre nach Iwasakis Entdeckung.

## Angepasster Schreibkopf

Das Schreibverfahren im PMR unterscheidet sich durch die veränderte räumliche Anordnung der magnetisierbaren Bereiche etwas vom herkömmlichen longitudinalen Speichervorgang. Der Schreibkopf muss nämlich einen viel fokussierteren magnetischen Fluss erzeugen, der tiefer in das Speichersubstrat hineinreicht. Dafür werden die Pole des Schreibkopfes so verändert, dass der Hauptpol schmaler und der Rückpol dafür umso breiter ausfällt.

Dadurch können am Hauptpol hohe Feldstärken erzeugt werden, die das Material ummagnetisieren und so Informationen sichern. Der Rückpol muss dagegen so breit sein, damit das Magnetfeld abgeschwächt wird und die bereits geschriebenen Daten nicht sofort wieder verloren gehen.

Dank dieses Designs des Schreibkopfes können die magnetischen Feldlinien senkrecht in das Speichersubstrat eindringen. Hinzu kommt noch ein weichmagnetisches Medium unter der senkrecht angeordneten Speicherschicht, die äußerst durchlässig für Magnetfelder ist und als magnetischer Spiegel für den Schreibkopf fungiert. Mit dieser Anordnung ist es möglich, die Feldlinien aus dem breiten Rückpol perpendikular in den Hauptpol zu leiten und so einen senkrecht ausgerichteten Speicherbereich zu erzeugen.

## Angepasster Lesekopf

Der Lesekopf bleibt beim PMR-prinzipiell gleich zum LMR-Verfahren, lediglich die Sensitivität wurde über den Riesenmagnetowiderstandseffekt erhöht. Bei diesem quantenmechanischen Effekt wird der elektrische Widerstand der Speicher-Bits gemessen: Je nach Orientierung der Speicherbits ist der Wert hoch (entgegengesetzt) oder gering (parallel).

## Wie Shingled Magnetic Recording (SMR) funktioniert

**Normalerweise speichern Festplatten ihre Daten in konzentrischen Ringen, die in einem gewissen Abstand voneinander liegen. Beim Shingled Magnetic Recording überlappen sich die Spuren jedoch wie Schindeln auf einem Dach („Shingle", engl. für „Schindel"), womit die Dichte angehoben werden kann. Damit ein gut gefülltes SMR-Laufwerk bei einem erneuten Schreibvorgang aber nicht sämtliche Spuren neu abspeichern muss, sind sie in Bändern organisiert, zwischen denen ein größerer Abstand liegt. So müssen maximal die Spuren eines Bandes neu geschrieben werden.**

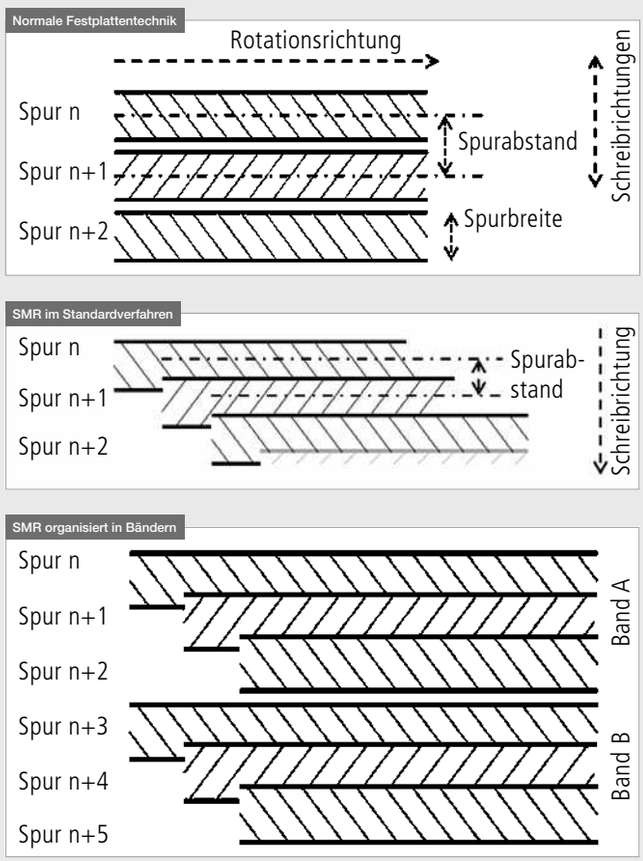



Bild: Tim Feldman, Garth Gibson

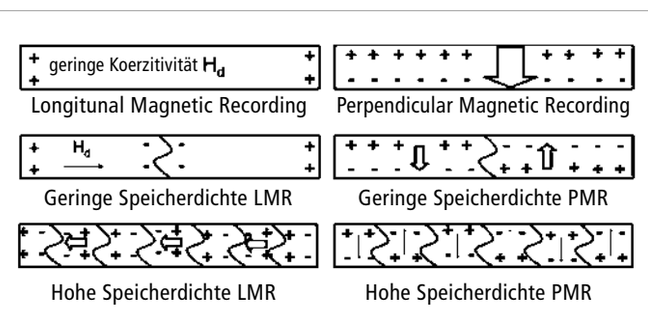
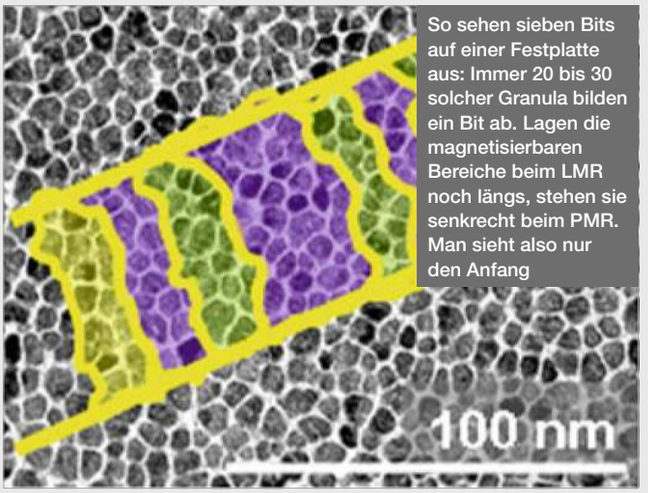


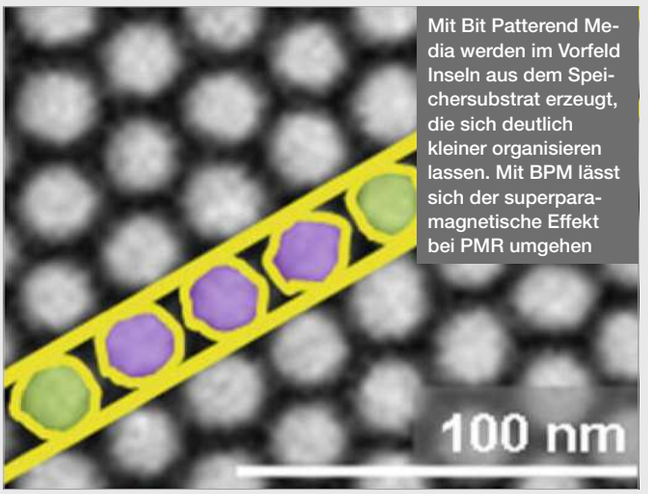

Bild: S. N. Piramanayagam, 2006

Hier wird nochmal klar, wie viel mehr Daten durch das PMR-Verfahren auf die gleiche Fläche eines Platters geschrieben werden können. Bei LMR ist der superparamagnetische Effekt der limitierende Faktor

## Wie Daten auf Festplatten physisch angeordnet sind




So sehen sieben Bits auf einer Festplatte aus: Immer 20 bis 30 solcher Granula bilden ein Bit ab. Lagen die magnetisierbaren Bereiche beim LMR noch längs, stehen sie senkrecht beim PMR. Man sieht also nur den Anfang




Mit Bit Patterend Media werden im Vorfeld Inseln aus dem Speichersubstrat erzeugt, die sich deutlich kleiner organisieren lassen. Mit BPM lässt sich der superparamagnetische Effekt bei PMR umgehen

Bild: O. Hellwig et al. 2013

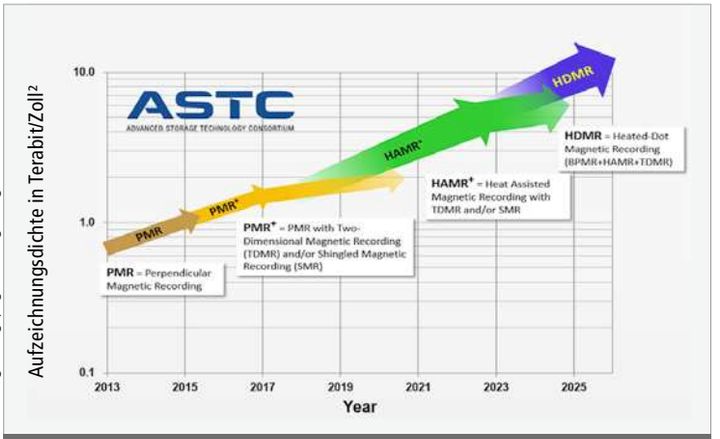



Bild: Seagte Blog, pcgh.de/seagteblog

2006 wurde LMR von PMR abgelöst. Noch 2019 soll HAMR kommen, welches dann mit Bit Patterend Media zu HDMR (Heat Dotted Magnetic Recording) wird

Je enger man die konzentrischen Speicherkreise auf einem Platter zieht, desto mehr Information kann man logischerweise auf selbigem speichern. PMR hat aber wie LMR praktisch schon die Obergrenze erreicht – mehr Daten pro Quadratzentimeter können in solchen Datenkreisen kaum gespeichert werden.

### Kleinere Bit-Größen

Wissenschaftler haben außerdem bereits die Bit-Größe – also die magnetisierbaren Körner, die für ein Bit stehen – bei PMR-Festplatten drastisch verringert. Dies konnte bewerkstelligt werden, indem Speichermaterialien herangezogen wurden, die eine hohe Koerzitivfeldstärke aufweisen. Damit wird die magnetische Feldstärke beschrieben, die benötigt wird, um eine ferromagnetische Substanz vollständig zu entmagnetisieren. Besitzt ein Material also eine hohe Koerzitivfeldstärke, kann es die Magnetisierung stabiler und effektiver halten – ideal also für Festplatten.

### Shingled Magnetic Recording (SMR)

Pfiffige Forscher haben beim Shingled Magnetic Recording (SMR) die Anordnung der magnetischen Spuren auf der Festplatte dahingehend optimiert, dass diese nun wie Dachschindeln angeordnet sind. Dadurch werden zwar benachbarte Bahnen teilweise überschrieben, diese werden dann aber neu geschrieben, um einen Datenverlust zu vermeiden. Außerdem werden diese „Datenschindeln“ in Gruppen organisiert, damit sich die Notwendigkeit des Neuschreibens nicht über den gesamten Platter zieht. Damit wird zwar teilweise die Geschwindigkeit verringert, dieses Verfahren ermöglicht aber eine Erhöhung der Datendichte um 25 Prozent bei gleicher Platterfläche.

### Heat Assisted Magnetic Recording (HAMR)

Ansätze wie SMR, heliumgefüllte Festplattengehäuse oder optimierte Lesealgorithmen können zwar kurzzeitig den Bedarf nach mehr „Fest“-Speicher stillen, über kurz oder lang kommen all diese Verfahren aber wieder an die Grenze ihrer Aufzeichnungsdichte. Erst das von Seagate entwickelte Heat Assisted Magnetic Recording (hitzeunterstütztes Schreibverfahren) erlaubt neue Speicherdichten über die bisherigen Limitierungen hinaus.

Bei HAMR wird der Bereich, in dem Daten geschrieben werden sollen, kurzzeitig mit einem Laser über die Curie-Temperatur des Platters aufgeheizt. Erst bei dieser Temperatur verlieren die Partikel ihre Magnetisierung und können einen neuen magnetischen Zustand annehmen. Dieser liegt auch nach dem Abkühlen stabil vor, obwohl die Speicherstrukturen bei diesem Verfahren noch kleiner dimensioniert sind als bisher. Damit kann der superparamagnetische Effekt, der LMR obsolet machte, für Festplatten mit HAMR umgangen werden – man fixiert die Magnetisierung gewissermaßen durch einen Laser.

Western Digital verfolgt mit MAMR (Microwave Assisted Magnetic Recording) einen anderen Ansatz, um die Limitierungen von PMR zu umgehen. Statt eines Lasers, der das Material aufheizt, erzeugt ein Spin-Torque-Oszillator ein Magnetfeld am Schreibkopf mittels Mikrowellen. Dadurch kommt man wieder in den Bereich von realisierbaren Feldstärken, die bei einer höheren Aufzeichnungsdichte notwendig werden. HAMR soll noch in diesem Jahr eine Festplatte mit 16 Terabyte erlauben, bis 2025 versprechen sich die Speicherforscher Festplatten mit 100 Terabyte und mehr.

### Bit Patterned Media (BPM)

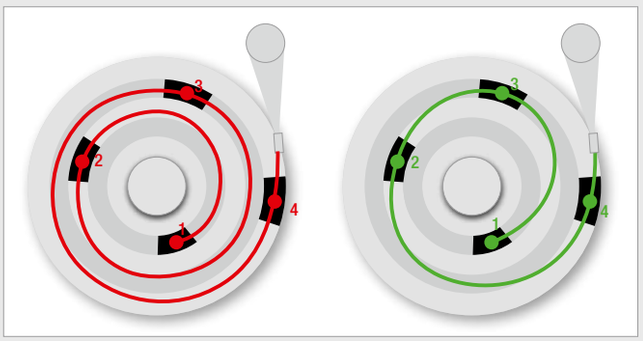
Für Speicher mit 100 Terabyte und mehr muss die HAMR-Technologie aber noch als „Bit Patterned Media“ organisiert werden. Dabei handelt es sich um magnetische Inseln, die jeweils im Nanometermaßstab Bits speichern (eine Insel, ein Bit). Bisher werden Daten auf einem magnetischen Film gespeichert, der eine granulare Struktur aufweist. Dort wird ein Bit mit ungefähr 20 bis 30 magnetisierbaren Körnern abgebildet (siehe Abbildung links).

Durch Herabsetzen der Durchschnittskorngröße kann man bis zum superparamagnetischen Effekt die Speicherdichte erhöhen.

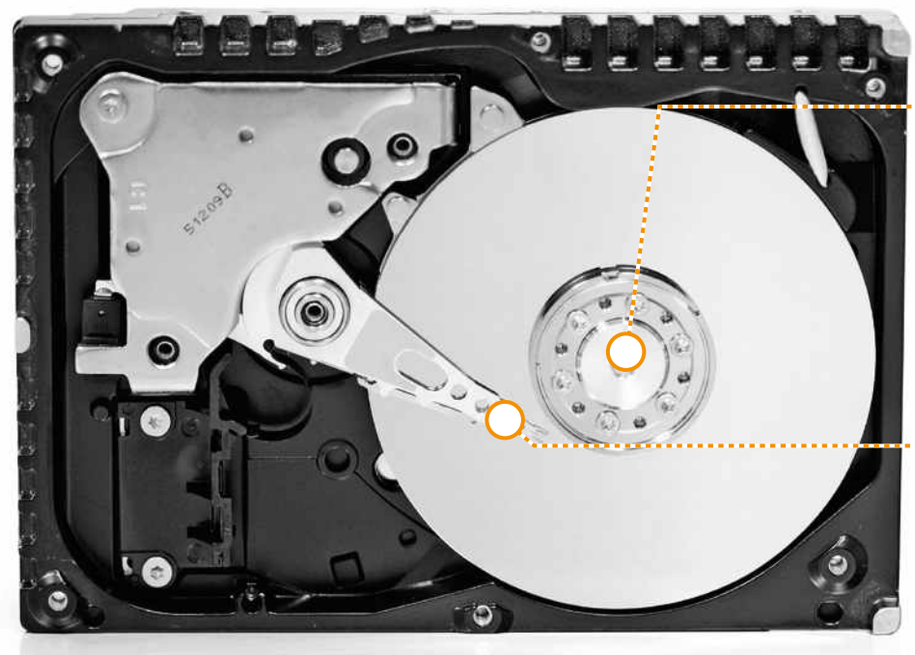
Bei BPM sind die Speicherstrukturen, in diesem Fall also die Inseln, durch Nanolithografie bereits vorgegeben. Jedes Bit wird bei BPM dann über eine starke Austauschwechselwirkung repräsentiert, anstatt wie beim granulosen Substrat durch viele schwache Wechselwirkungen. Das bedeutet im Umkehrschluss, dass die Energiebarriere nun proportional zum Inselvolumen ist und nicht mehr zum Volumen einzelner Speicherkörner. Damit kann die Dichte erneut angehoben werden, denkbar sind bis zu 300 Terabyte pro Quadratzoll.

## Wie Schreibvorgänge mit NCQ optimiert werden

**Die Umsortierung der Lese-/Schreibzugriffe berücksichtigt nicht nur die hier dargestellte Kopfbewegung, sondern auch die Rotationslatenz bei der Optimierungsstrategie.**




## Wie eine Festplatte intern aufgebaut ist



### Spindelmotor

Je nach Modell treibt der Motor die Platter auf bis zu 15.000 U/min – üblich sind 5.400 bis 7.200 U/min.

### Aktuator/Leseköpfe

Jeder Lesekopf am präzise gesteuerten Aktuator hängt nur den Bruchteil eines Millimeters über dem Platter.

### Platter 1

Die Platter tragen außen eine ferromagnetische Schicht, deren einzelne Elemente die Bits rerpäsentieren.

### Controller 2

Ein meist ARM-basierter Prozessor mit DRAM-Cache optimiert und organisiert die Lese-/Schreiboperationen.

# So funktioniert die Infrastruktur: Netzteil

Bitte beachten Sie: Öffnen Sie nie ein Netzteil - im Gerät gibt es keine zu wartenden Teile! Die Bauteile können elektrische Ladung speichern, der bei Kontakt zum Stromtod führen kann!

Mit den 230 Volt Wechselspannung aus europäischen Steckdosen können hochempfindliche Bauteile nicht arbeiten. Für den Computer sind daher 3,3 bis 12 Volt Gleichspannung (DC) notwendig, um einen einwandfreien Betrieb zu garantieren. Wie wird dieser Strom gewandelt und warum ist dieser Prozess überhaupt wichtig?

## Schaltnetzteildesign

Die heutige Stromversorgung in Rechnern übernehmen Schaltnetzteile, sodass die Energieübertragung üblicherweise mit Frequenzen von 15 kHz bis 300 kHz vonstattengeht. Natürlich gibt es noch weitere Netzteiltypen wie etwa ein Kondensator- oder ein Trafonetzteil, die ein anderes Design verwenden, diese sind aber nicht in der Lage, die Anforderungen der PC-Hardware adäquat zu erfüllen.

Beim Schaltnetzteil wird der Strom aus der Dose im Spannungswandler durch Transistoren in kleine Pakete zerlegt. Transistoren fungieren dabei als An/Aus-Schalter (siehe CPU-Kapitel). Entsprechend der verwendeten Frequenz werden die Strompakete dank anderer Komponenten wie Kondensatoren oder Spulen weitergegeben. Bevor aber irgendwelche Computer-Hardware die Strompakete bekommen kann, werden diese wieder zu einem kontinuierlichen Strom zusammengefügt und können dann problemlos verbraucht werden.

Das hat gegenüber linear-regulierten Netzteilen einige Vorteile, beispielsweise ein geringeres Gewicht. Dieses kann nämlich bei hohen Frequenzen reduziert werden, indem sowohl beim Ferrit- oder Eisenkern als auch bei den Kupferwicklungen des Transformators die Masse verringert wird. Ein Umspanner, der für eine Leistung von 4.000 Watt geeignet sein soll, wiegt bei einer Frequenz von 50 Hz – was übrigens der europäischen Netzspannung entspricht – ganze vier (!) Kilogramm. Erhöht man dagegen die Schaltfrequenz auf 125 kHz, sind es nur noch knapp 470 Gramm, was einer Gewichtseinsparung von 88,25 Prozent entspricht.

## Wirkungsgrad

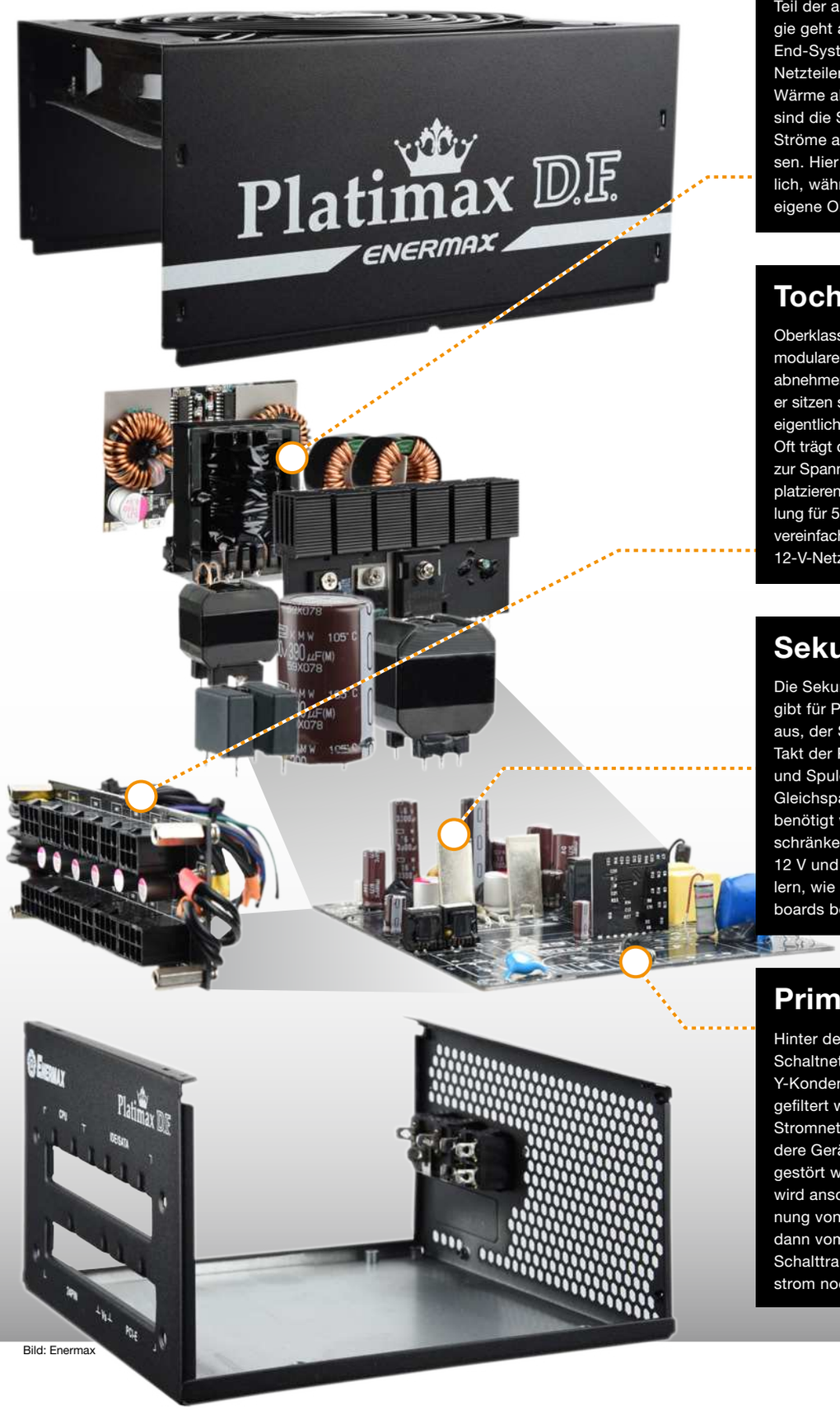
Ein anderer wichtiger Aspekt von Schaltnetzteilen ist der erhöhte Wirkungsgrad. Bei linear-regulierten Spannungswandlern ist die Effizienz maßgeblich vom Unterschied zwischen Input- und Output-Stromspannung abhängig. Hier wird die ausgegebene Spannung über die Verlustleistung in Form von Wärme reguliert, sodass Wirkungsgrade von 30 bis 40 Prozent normal sind. Bei Schaltnetzteilen beträgt die Effizienz dagegen schon 60 bis 70 Prozent, weil Schaltverluste an den Transistoren entstehen. Dies kann aber durch eine Schaltoptimierung verhindert werden. Netzteile mit einem guten Design, einer angepassten Frequenz und dem Zero-Current- (ZCS) respektive dem Zero-Voltage-Switching (ZVS) können Wirkungsgrade von bis zu 95 Prozent erreichen.

Die beiden genannten Schaltmethoden werden über einen LCC-Resonanzwandler realisiert, sodass die Schaltverluste an den Transistoren und die Verlustleistung minimiert werden können. Ist ein solcher Transistor geöffnet, fließt ein hoher Strom, aber keine Spannung. Im geschlossenen Zustand liegt stattdessen eine hohe Spannung an, aber es fließt kein Strom. Da der

## Netzteil-Effizienz der einzelnen 80-Plus-Standards

| Effizienzstufen | 10 Prozent | 20 Prozent | 50 Prozent | 100 Prozent |
|---|---|---|---|---|
| 80 plus | Nicht definiert | 82% | 85% | 82% |
| 80 plus Bronze | Nicht definiert | 85% | 88% | 85% |
| 80 Plus Silber | Nicht definiert | 87% | 90% | 87% |
| 80 plus Gold | Nicht definiert | 90% | 92% | 89% |
| 80 plus Platinum | Nicht definiert | 92% | 94% | 90% |
| 80 plus Titanium | 90% | 94% | 96% | 94% |

# Netzteil-Aufbau




## Kühlung

Die Effizienz von Netzteilen ist endlich – ein Teil der aufgenommenen elektrischen Energie geht als Abwärme verloren. Bei High-End-Systemen sind auch mit hocheffizienten Netzteilen schnell einige Dutzend Watt Wärme abzuführen. Die kritischsten Bauteile sind die Schalttransistoren, die große Ströme auf kleinem Raum bewältigen müssen. Hier sind zwingend Kühlkörper erforderlich, während anderen Bauteilen meist ihre eigene Oberfläche zur Wärmeabgabe reicht.

## Tochterplatine

Oberklassenetzteile verfügen meist über ein modulares Kabelsystem, bei dem sich Kabel abnehmen lassen. Die entsprechenden Stecker sitzen seitlich im Netzteil auf einer vom eigentlichen Schaltnetzteil getrennten Platine. Oft trägt diese auch weitere Kondensatoren zur Spannungsglättung. Einige Hersteller platzieren gar die gesamte Abwärtswandlung für 5 V und 3,3 V an dieser Stelle. Das vereinfacht die primäre Platine zu einem reinen 12-V-Netzteil, erschwert aber die Kühlung.

## Sekundärseite

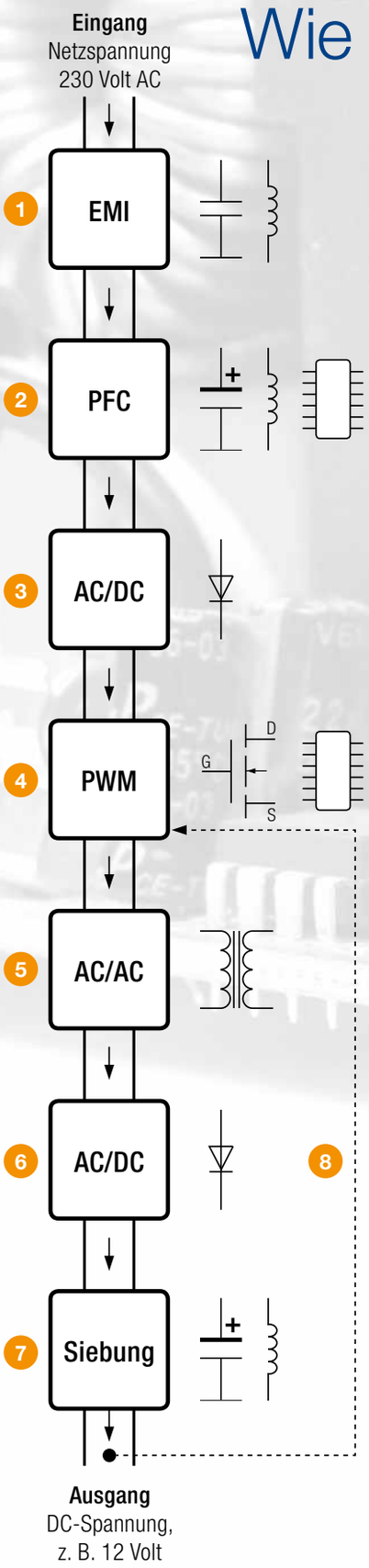
Die Sekundärwicklung des Transformators gibt für PCs typische, niedrige Spannungen aus, der Strom ist aber immer noch mit dem Takt der Primärseite gepulst. Kondensatoren und Spulen machen daraus eine saubere Gleichspannung, wie sie von PC-Hardware benötigt wird. Moderne DC-DC-Layouts beschränken sich auf die Transformation von 12 V und erzeugen daraus mit Abwärtswandlern, wie sie von Grafikkarten und Mainboards bekannt sind, auch 3,3 V und 5 V.

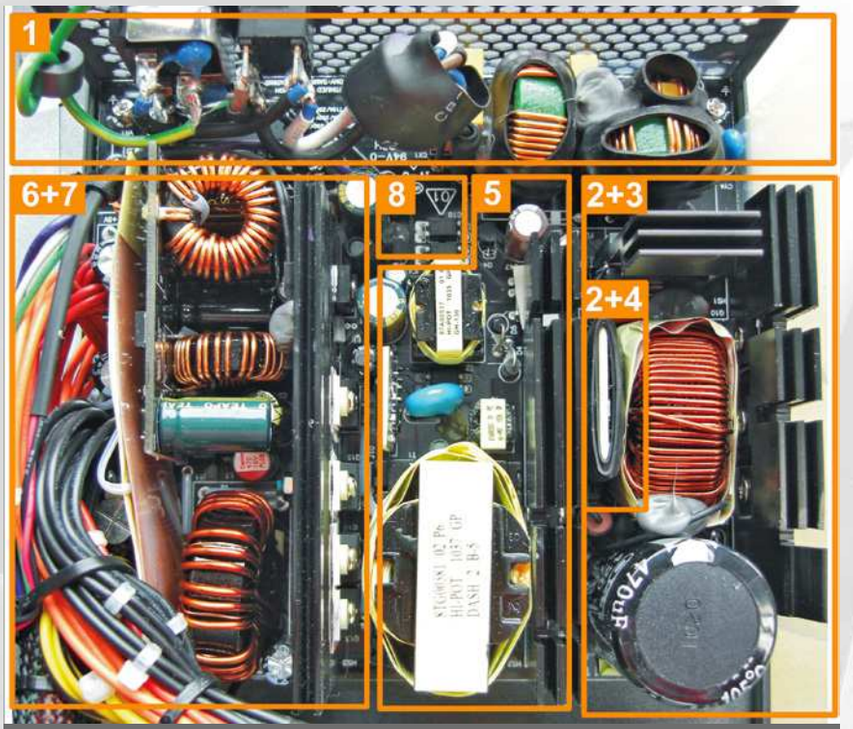
## Primärseite

Hinter dem Kaltgerätestecker eines Schaltnetzteils dienen Spulen und X- bzw. Y-Kondensatoren als Eingangsfilter. Ausgefiltert werden Unsauberkeiten aus dem Stromnetz und es wird verhindert, dass andere Geräte durch die Hochfrequenztechnik gestört werden. Der bereinigte Wechselstrom wird anschließend in in eine Gleichspannung von 300 bis 400 V umgewandelt, die dann vom Primärkondensator geglättet wird. Schalttransistoren zerhacken diesen Gleichstrom noch in einen gepulsten Strom.

Bild: Enermax

# Wie ein Netzteil den Strom wandelt




In diesem Bild sehen Sie, wie die einzelnen Schaltungen in einem Netzteil realisiert werden. Teilweise gehen die Schaltungen fließend ineinander über und sind nur schwer auseinanderzuhalten.

**1 EMI**
Der EMC-/EMI-Filter sorgt für die Störimmunität gegenüber dem Hausstromnetz.

**2 PFC**
Die Power Factor Correction dient zur Blindleistungskompensation und erhöht unter anderem die Effizienz.

**3 AC/DC**
Die 230-Volt-Netzspannung wird gleichgerichtet.

**4 PWM**
Damit der kompakte Tansformator arbeiten kann, wird per Pulsweitenmodulation eine hochfrequente Spannung erzeugt. Der Taktgeber ist ein Schalter oder „Zerhacker".

**5 AC/AC**
Der Transformer sorgt für die Übertragung beziehungsweise Transformation der Netzspannung in die geringere Spannung.

**6 AC/DC:**
Die heruntertransformierte Spannung wird erneut gleichrichtet.

**7 Siebung**
Die Spannung wird geglättet und gesiebt und anschließend den PC-Bauteilen zugeführt. Zudem werden per DC/DC-Wandlung die 3,3 sowie 5 Volt erzeugt.

**8 Feedback**
Dient zur Überwachung der Ausgangsspannung und zur Regelung des PWM-Moduls

Schaltvorgang etwas Zeit benötigt, entsteht eine Verzögerung, in der sowohl Spannung als auch Strom anliegen. Durch Designoptimierungen und das genaue Schalten des LLC-Resonanzwandlers, wenn der Strom sich am Nulldurchgang befindet, werden die Schaltverluste ebenso Null. Dadurch erhöht sich die Leistungsdichte und damit auch der Wirkungsgrad des Netzteils. Erreicht werden kann dies mit einem sinusförmigen Signal.

Schaltnetzteile sind in der Regel aus mehreren Bereichen aufgebaut, die unterschiedliche Aufgaben bei der Stromumwandlung und der Stromqualität erledigen. Insgesamt kann man deshalb den Grundaufbau in mehrere Stationen zerlegen (siehe Blockdiagramm links), die wir im Folgenden erklären.

## Eingangsfilterung – EMI

Schaltnetzteile haben nicht nur Vorteile wie eine bessere Effizienz, sondern erzeugen durch die Schaltvorgänge der MOSFETs ungewollte elektromagnetische Interferenzen (EMI, auch als Funkstörungen bekannt), die andere elektronische Geräte im Netz beeinflussen können. Jeder kennt sicherlich die elektromagnetischen Interferenzen, die entstehen, wenn ein Handy in die Nähe eines Lautsprechers kommt. Auf der anderen Seite muss das Netzteil – und damit der Computer – vor Störungen aus dem Stromnetz geschützt werden.

Mittels der EMI-Filterstufe direkt nach dem Kaltgerätesteckereingang wird diese wichtige Station erreicht. Die Filterung erfolgt immer vor der Gleichrichtung des Wechselstroms, da an dieser Position Störungen der Netzteil-Dioden (für die Spannungsstabilität zuständig) entstehen und so mit herausgefiltert werden müssen. Um die Störungen zu unterdrücken, werden unter anderem X- und Y-Kondensatoren (Entstörkondensatoren) eingesetzt. Diese Baugruppen leiten hochfrequente Störsignale, die durch den Betrieb von Elektrogeräten entstehen, gegen die Masse (X-Kondensator) oder den Neutralleiter (Y-Kondensator), sodass die ungewollten Störungen reduziert werden.

Daneben werden auf der Filterstufe noch Spulen, Metall-Oxid-Varistoren (MOV, variabler Resistor) und Sicherungen genutzt. MOVs sind spannungsabhängige Widerstände, die das System vor Spannungsspitzen aus dem Versorgungsnetz schützen. Billige Netzteile, die auf solch eine MOV verzichten, sollten Sie dann nur mit einem zugeschalteten Überspannungsschutz betreiben, sonst könnte solch ein Spannungsausschlag die Hardware beschädigen. In diesem Bereich kommen auch Spulen zum Einsatz, da diese mit steigender Frequenz ihren Innenwiderstand erhöhen und so hochfrequente Störsignale filtern.

Direkt hinter der EMI-Filterung wird meist ein Heißleiter oder ein Relais eingesetzt, damit der Einschaltstrom möglichst gering bleibt. Dieser kann nämlich bis zu zehn Mal so ausgeprägt sein wie der Nennstrom und dementsprechend zu Problemen mit den Stromsicherungen im Haus führen.

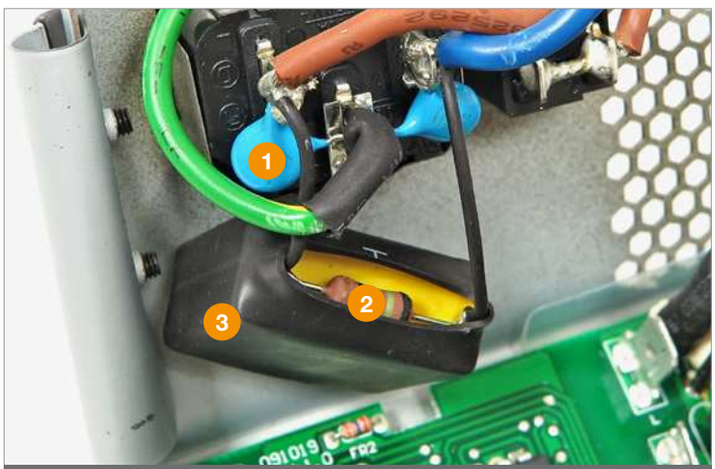
## Leistungskorrekturfilter – PFC

Bei der Versorgung des Schaltnetzteils mit der sinusförmigen Wechselspannung aus dem Netz entstehen phasenverschobene Eingangsströme, da die Spannungswandler den Strom unsymmetrisch entnehmen. Diese negativen Effekte sind nicht gewollt und bestehen aus Oberschwingungen, die hochfrequente Anteile innehaben und so in anderen Geräten im Netz Störungen verursachen können. Vermieden werden kann dies durch einen Leistungsfaktor, der möglichst nahe bei 1 liegt.

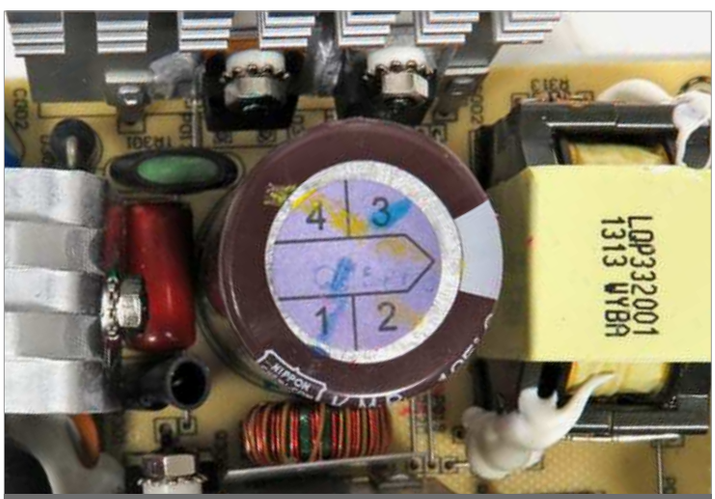
Diese Kenngröße beschreibt das Verhältnis von Wirkleistung (tatsächlich verwendbare Energie) und Scheinleistung (gesamte zugeführte Energie aus Wirkleistung und Blindleistung). Da der Leistungsfaktor nicht immer bei 1 liegt, erhöht der Leistungskorrekturfilter (PFC, Power Factor Correction oder Compensation) den Wert.

Bei kleinen Leistungen bis 200 Watt erfolgt dies teilweise noch passiv, in modernen Schaltnetzteilen liegt aber in der Regel eine aktive elektronische Schaltung (APFC) vor, die den Leistungsfaktor kontrolliert. Die APFC ist quasi ein zusätzliches Schaltnetzteil, welches dem eigentlichen Spannungswandler vorgeschaltet wird und den bereitgestellten Strom durch ein PWM-Signal kontrolliert.

Diese Schaltung sorgt dafür, dass der aufgenommene Strom immer



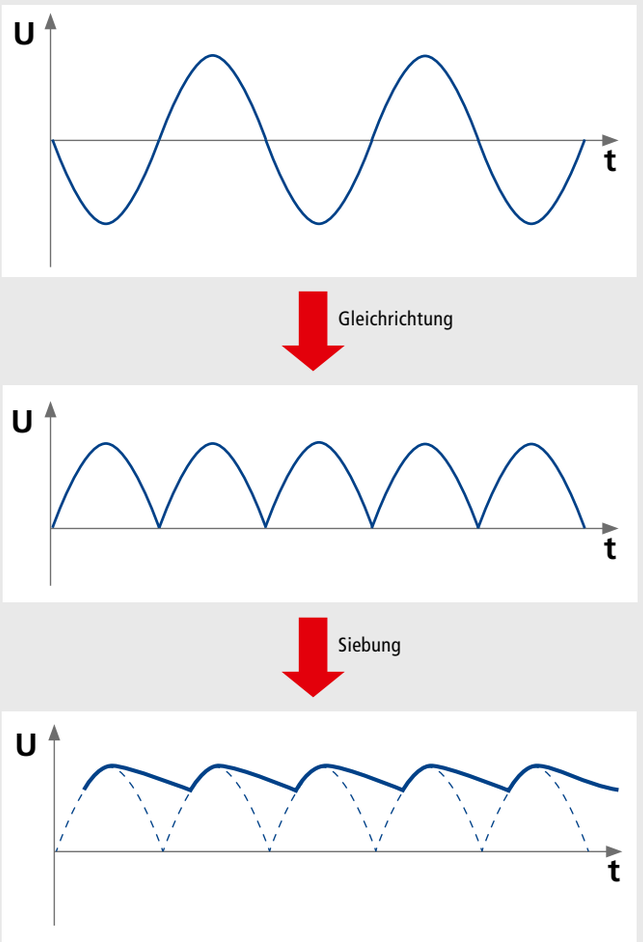

Direkt nach dem Kaltgerätestecker kommt die Eingangsfilterung, die Störsignale aus dem Netz filtern beziehungsweise das Netz vor ebensolchen aus dem Spannungswandler schützt. (1) Y-Kondensator, (2) Ableitwiderstand, (3) X-Kondensator




Hier sehen Sie den aktiven Leistungskorrekturfilter, der direkt vor dem Siebkondensator eine Spule verbaut hat, die den Einschaltstrom ohne Dissipation kompensieren kann

## Info: Wechselstrom in Gleichstrom umwandeln

**Sie sehen eine vereinfachte schematische Darstellung, wie aus Wechselstrom computertauglicher Gleichstrom wird. Wechselstrom wird gleichgerichtet und dann gesiebt.**




der sinusförmigen Netzspannung entspricht und fährt diese gegebenfalls nach, sollte sie von einer Idealschwingung abweichen. Die APFC besteht deshalb in der Regel aus einem Gleichrichter mit direkt nachgeschaltetem Aufwärtswandler.

Zunächst löst das PWM-Signal die Transistoren der APFC aus, welche den erzeugten gleichgerichteten Strom in konstante Pulse zerhackt und die Spannung dann auf 350 bis 400 Volt anhebt. Damit wird auch der Primärkondensator aufgeladen. Dieser siebt die Wechselspannung und versorgt das nachgeschaltete Netzteil mit Energie. So lassen sich mitunter Schwankungen im Netz ausgleichen, es entstehen dadurch aber auch neue hochfrequente Störungen (da ja ein zusätzliches Schaltnetzteil eingebracht wird), die mittels Netzfilter unterdrückt werden müssen.

### Primäre Gleichrichtung

Computer nutzen keinen Wechselstrom, der aus der Steckdose kommt, sondern Gleichstrom. Um diesen zu erzeugen, muss der Netzstrom gleichgerichtet werden. Der Strom aus der Steckdose wechselt seine Polung (Richtung) in regelmäßiger Wiederholung und der Spannungsverlauf entspricht einer Sinuskurve. Bei der Gleichrichtung bekommen alle Amplituden, die eine negative Polung aufweisen (also die Täler der Sinusschwingung im Graph), ein positives Vorzeichen, sodass die Täler der Sinuskurve in einem Graph vollständig nach oben geklappt werden.

Die Topologie, die dafür heutzutage verantwortlich ist, nennt sich Gleichrichterbrücke und je nachdem, wie viele MOSFETs eingesetzt werden, spricht man entweder von einer sogenannten Half oder Full Bridge. Die MOSFETs müssen aber aktiv angesteuert werden, damit die Netzspannung in eine Gleichspannung gewandelt werden kann.

Früher wurde dies ohne Ansteuerung über eine Diodenbrücke realisiert, die den Wechselstrom während der positiven Phase der Sinuskurve („Berge“ der Schwingung) in eine Richtung durchlässt (Durchlasspolung). Der Nebenwiderstand der Diode kehrt dann den negativen Teil der Sinusschwingung („Täler“) um (Sperrpolung), sodass der Strom nun keine Halbschwingung mehr besitzt. Bevor aber der Strom Computer-Hardware betreiben kann, müssen die Elektronen noch weitere Stationen durchlaufen, denn im Moment liegt noch eine gepulste Gleichspannung vor.

### Transformator

Der gepulste gleichgerichetete Gleichstrom aus dem Primärkondensator wird nun durch einen Transistor in ein rechteckiges Wechselstromsignal gewandelt, das eine Frequenz von 50 bis 125 kHz aufweist und in den Transformator eingespeist wird. Wie erwähnt ist die Frequenz notwendig, damit Gewicht und Größe des Transformators im Rahmen eines ATX-Netzteils bleiben können. Die Stromkreise beider Wicklungen sind durch eine galvanische Trennung voneinander isoliert, sodass der angelegte hochfrequente Wechselstrom an der Primärspule ein magnetisches Feld erzeugt, welches wiederum Strom in der Sekundärspule induziert. Der Umspanner wandelt also über zwei oder mehrere Kufperspulen, die über einen Eisen- oder Ferritkern miteinander verbunden sind und unterschiedlich starke Wicklungen aufweisen, die Eingangsspannung in die benötigten 12 Volt um.

### Sekundäre Gleichrichtung

Nachdem der Strom die Transformation durchlaufen hat und damit

auf der Sekundärseite des Netzteils angelangt ist, muss der nun heruntergewandelte Wechselstrom in computertauglichen Gleichstrom gerichtet werden. Dies kann passiv über Schottky-Dioden erfolgen oder synchron mittels eines sogenannten Synchronous Rectifier, der anstelle von Dioden mehrere MOSFETs verwendet. Auch hier lässt sich wieder an der Effizienzschraube des Netzteils drehen, da Schottky-Dioden einen typischen Spannungsabfall besitzen, der Leistungsverluste nach sich zieht.

Solch ein typischer Spannungsdrop beträgt beispielsweise 0,5 V. Wenn wir nun 28 A durchleiten wollen, dann erzeugt die Schottky-Diode 14 Watt Dissipation (28 A × 0,5V). Bei MOSFETs ist dieser Wert viel geringer, da diese Transistoren über einen niederohmigen Einschaltwiderstand von ca. 1 mΩ verfügen. In unserem Fall bedeutet das, dass ein MOSFET 0,784 Watt dissipiert und damit die Effizienz um 5,6 Prozent gesteigert wird.

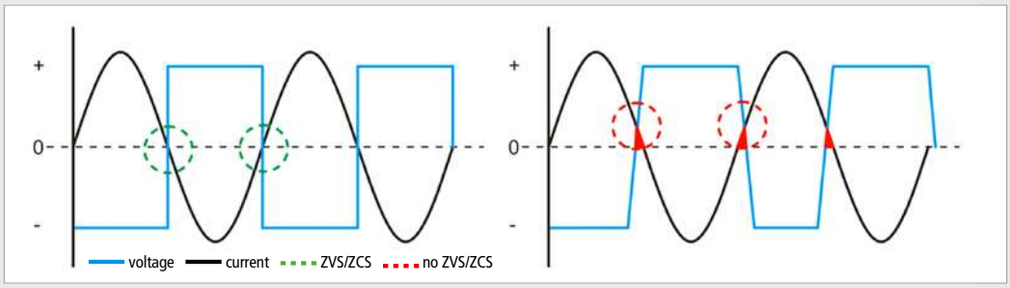
Neben diesen beiden Typen der Gleichrichtung gibt es auch noch Hybridschaltungen aus MOSFETs und Schottky-Dioden, um so die Kosten bei einer gleichzeitigen Steigerung der Effizienz (verglichen mit einer passiven Lösung) zu drücken.

### Entstehen der Spannungen

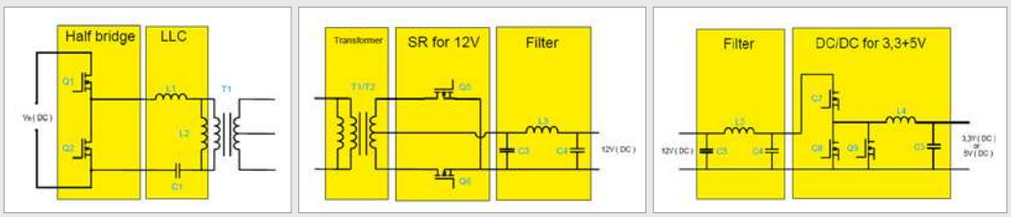
Die Anforderungen an die unterschiedlichen Schienen im Netzteil haben sich in den letzten Jahren durch moderne Hardware drastisch geändert: Durch leistungsfähigere Grafikkarten und Prozessoren wird der 12-Volt-Schiene immer mehr Bedeutung zuteil. Um die gewünschten Spannungen in einem Netzteil zu erzeugen, werden die Schienen entweder in Gruppen reguliert, durch eine Spannungsregulation

## Wie Netzteil-Hersteller den Wirkungsgrad immer weiter erhöhen

**Die hier abgebildeten Topologien ermöglichten es Netzteilherstellern, in den letzten Jahren, den Wirkungsgrad immer weiter gen Maximum zu verschieben.**

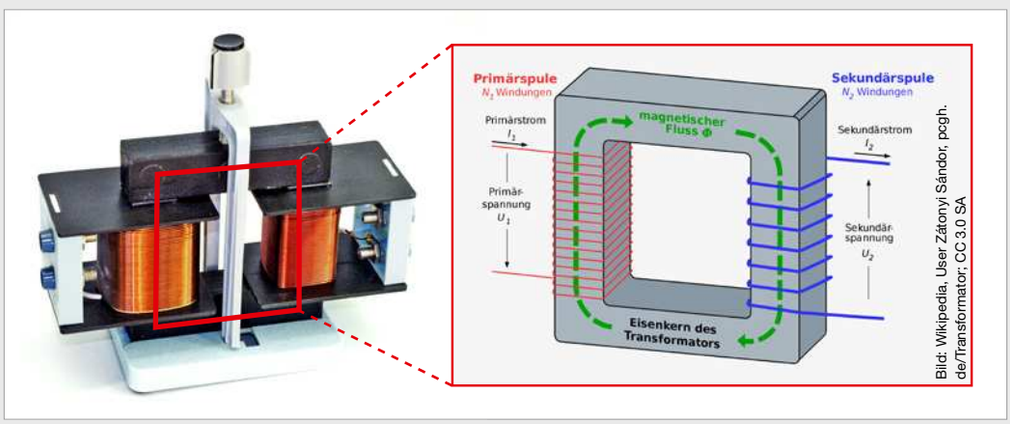



**Durch Reduzierung der Schaltverluste (Grafik oben, LLC), synchrone Gleichrichter (SR) und die DC-DC-Wandler erreichen High-End-Netzteile heutzutage Effizienzwerte von bis zu 95 Prozent. In Zukunft werden wahrscheinlich andere Materialien wie Galliumnitrid, das im Gegensatz zu Silizium einen geringeren Widerstand aufweist und so eine weitere Verringerung von Schaltverlusten an den FETs erlaubt. Damit wäre auch einer Erhöhung der Leistungsdichte um bis zu 40 Prozent möglich.**




## Wie Transformatoren intern Spannungen umwandeln

**Transformatoren bestehen aus zwei Spulen mit jeweils einer unterschiedlich starken Wicklung aus einem Kupferdraht. Beide Wicklungen teilen sich einen gemeinsamen Ferrit- oder Eisenkern und sind so über diesen „magnetisch" verbunden. Die Eingangsspannung an einer der Spulen wird dann in eine veränderte Ausgangsspannung an der anderen Spule umgewandelt. Dabei entspricht das Verhältnis der Wicklungen zueinander dem Verhältnis der Spannungen.**



# Netzteil-Komponenten in der Übersicht

**In einem Netzteil kommen unterschiedliche elektronische Bauteile zum Einsatz. Die folgende Übersicht zeigt, welche das sind und wozu diese benötigt werden.**

## Diode

Die Diode ist ein Halbleiterbauelement, welches Strom nur in einer bestimmten Richtung durchlässt. Im Netzteil kommen leistungsfähige Schottky-Dioden zum Einsatz, um die Wechselspannung gleichzurichten. Diese Art Diode besitzt einen Metall-Halbleiter-Übergang und eignet sich besonders für hochfrequente Spannungen und starke Ströme.

## Kondensator

Ein Kondensator kann Energie kurzzeitig speichern; dies wird als elektrische Kapazität bezeichnet. Im Bild ist ein Elektrolytkondensator (kurz Elko) mit einer Kapazität von 390 Mikrofarad zu sehen. Dieser Kondensator dient zum Glätten beziehungsweise Sieben der gleichgerichteten Wechselspannung und schwächt Restschwingungen ab.

## Transformator

Ein Transformator besteht aus mindestens zwei Spulen und einem Ferrit- oder Eisenkern. In die sogenannte Primärspule wird die höhere Wechselspannung eingespeist, sodass in der Sekundärspule niedrigere Spannung induziert wird. In der Regel sind die beiden Spulen isoliert, sodass eine galvanische Trennung vorliegt.

## Mikrocontroller

Ein Mikrocontroller steuert das „Zerhacken" der Gleichspannung und die Leistungsfaktorkorrektur. Die Mikrocontroller auf der Zusatzplatine übernimmt gleich zwei Funktionen: Zum einen steuert er die Leistungsfaktorkorrektur für eine bessere Effizienz und zum anderen erzeugt er den Takt für das Zerhacken der Gleichspannung.

## Spule

Eine Drosselspule wird hauptsächlich zur Unterdrückung von Störimpulsen eingesetzt. Eine Spule besteht aus einem magnetisierbaren Kern, der mit Draht umwickelt ist. Der Strom durchfließt die Spule und durch die Selbstinduktion werden hochfrequente Wechselstromamplituden oder Störstrahlungen gedämpft beziehungsweise abgeschwächt.

## Brücke

Die DC-DC-Brücke im Netzteil reduziert 12 Volt Gleichspannung auf die zusätzlich notwendigen 5 und 3,3 Volt. Auf dem Bauteil kommen ein Mikrocontroller, Spulen und Kondensatoren zum Einsatz. Dank dieser Technik sind Netzteile mit 90 Prozent Effizienz überhaupt erst möglich. Allerdings wird die Baugruppe nicht bei allen Netzteilen als zusätzliche Platine ausgeführt.

Bilder: Be quiet, Infineon

angepasst oder jede Leitung wird unabhängig voneinander erzeugt.

Bei älteren gruppenregulierten Geräten wurden mindestens zwei Ausgangsspannungen über denselben Quelltransformator generiert. Dieser besaß dann zwei unterschiedlich gewickelte Sekundärspulen und konnte so zwei Spannungen ausgeben. Problematisch wird dieses Design, wenn das Netzteil eine der beiden Spannungen korrigieren muss, da ein Anpassen der Voltzahl an der Primärspule immer eine Veränderung in beiden Spulen nach sich zieht.

Sind also mehrere Ausgangsspannungen abhängig von einem Transformator, werden diese deshalb auch in Mitleidenschaft gezogen. Werden beispielsweise 12 und 5 Volt aus einem Transformator generiert und es kommt zu einem Lastungleichgewicht, weil auf der 12-Volt-Schiene ein größerer Verbraucher zugegen ist als auf der 5-Volt-Schiene, dann hat der Controller Probleme damit, eine saubere Spannungsregulation hinzubekommen. Obwohl das Netzteil versucht, nur die 12-Volt-Schiene mit mehr Energie zu versorgen, wird automatisch die 5-Volt-Leitung mit angehoben, da beide Spannungen über einen Transformator aus dem Versorgungsstrom gebildet werden. Bei solchen Crossload-Szenarien schaffen es gruppenregulierte Netzteile deshalb auch nicht, unterhalb der 5-Prozent-Toleranzgrenze zu bleiben, die die ATX-Norm vorgibt.

Dagegen werden alle Spannungen bei unabhängig regulierten Netzteilen getrennt voneinander erzeugt. Jede Schiene bekommt in solch einem Netzteil ihre eigene unabhängige Schaltung, sodass es nicht zu Spannungsabweichungen oder -abfällen kommt. Dieses Design findet man hauptsächlich in Enthusiasten-Spannungswandlern, bei denen der Preis eine untergeordnete Rolle spielt.

Das wohl heutzutage am häufigsten eingesetzte Netzteildesign ist aber die Spannungsregulation, auch als DC-DC-Wandler bekannt. Zunächst wird aus der Netzspannung über die bereits erwähnten Stationen in einem Netzteil eine saubere Ausgangsspannung von 12 Volt erzeugt. Damit aber die beiden anderen Spannungen generiert werden können, kommen sogenannte Buck Converter zum Einsatz. Diese Baugruppen wandeln die 12 Volt dann auf die angestrebten 3,3 beziehungsweise 5 Volt ab und nehmen so direkt die erzeugten Ausgangsspannung des Quelltransformators ab. Durch Controller werden die beiden geringen Spannungen überprüft und eventuell nachgeregelt, sodass durch diesen Aufbau im Prinzip kein Einfluss mehr auf die 12-Volt-Schiene erfolgt.

Die Spannungen -12 Volt und 5VSB (Standby) fehlen in dieser Übersicht noch. Die -12-Volt-Leitung ist üblicherweise über eine konventionelle Diode realisiert, da meistens weniger als 1 A für das Netzteil beziehungsweise den Rechner benötigt wird. Bei der 5-Volt-Standby-Leitung wird dagegen normalerweise auf eine vollständig eigene Verschaltung gesetzt, die auch über einen eigenen Transformator verfügt, da diese Spannung kontinuierlich ausgegeben wird, selbst im Standby des Netzteils.

Wir haben nun also den Werdegang der Elektronen durch einen Spannungswandler verfolgt und beobachtet, wie aus alternierendem Wechselstrom mit einer Spannung von 230 Volt computertaugliche 3,3, 5 und 12 Volt Gleichspannung wurden. Auf diesem Weg wurde der Strom nicht nur umgewandelt, son-

dern auch gesäubert und geglättet, damit er die Anforderungen an die sensible Hardware erfüllt.

## Top-Topologie LLC-Wandler

Um die Effizienz zu steigern und damit sich die ausgegebenen Spannungen immer mehr dem Ideal annähern, gibt es neben den bereits angesprochenen Topologien noch weitere Kniffe.

Eine wichtige Schaltung, die auch bereits erwähnt wurde, ist der LLC-Resonanzwandler. Dieser findet in nahezu allen Titanium- und Platinum-Netzteilen Anwendung, da man durch das ZVS die Schaltverluste minieren kann. Dieser Wandler, bestehend aus zwei Spulen (L) und einem Kondensator (C), wird innerhalb der primären Gleichrichtung eingesetzt und direkt hinter eine Half oder Full Bridge gesetzt.

Diese Halb- oder Vollbrückendurchflusswandler haben heutzutage ihr Optimierungsmaximum mit den eingesetzten Frequenzen von 50 bis 125 kHz erreicht. Noch höhere Schwingungszahlen würden wieder Probleme einführen, wie etwa Verluste durch die Ummagnetisierung des Transformatorkerns. Hinzu kommt ein linearer Anstieg der Verlustleistung an der MOSFET, da die Schaltgeschwindigkeit zwischen Off und On durch physikalische Grenzen beschränkt ist.

Um dennoch die Effizienz zu erhöhen, kommt der nachgeschaltete LLC-Resonanzwandler zum Einsatz. Auf der Primärseite wird nun ein Schwingkreis mit dem Transformator und dem zusätzlichen Kondensator erzeugt, sodass anstatt der Rechteckimpulse eine sinusförmige Welle entsteht. Dieser Schwingkreis gibt dann auch die Frequenz vor, sodass zwei MOSFETs nur noch die Energie liefern müssen. Erst dadurch ist das Zero-Voltage-Switching überhaupt möglich.

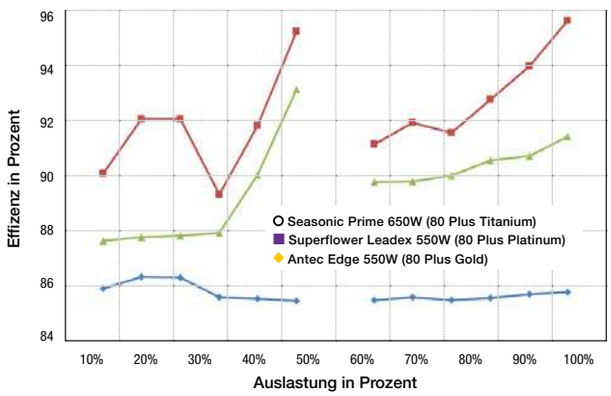
Der LLC-Resonanzwandler findet schon seit geraumer Zeit Anwendung, beispielsweise in Geräten, die keine dynamischen Lasten benötigen. Ältere Netzteile ohne LLC-Schaltung reagierten auf Laständerung mit der Anpassung der Frequenz und/oder Pulsbreite. Steigt die Last nun in einem Netzteil mit LLC-Resonanzwandler, kann eine Akkodomation nur über die Pulsbreite erfolgen, da der Schwingkreis eine eigene Frequenz vorgibt. Dies ist aber problematisch, da so nicht mehr adäquat auf veränderte Lasten reagiert werden kann. Eigentlich würde sogar bei steigender Belastung die Frequenz ansteigen, da der Transformator Teil des Kreises ist und die Induktivität bei steigender Belastung abnimmt.

Um dies zu umgehen, haben die Ingenieure von Seasonic damals einfach eine weitere Spule in den Schwingkreis eingeführt, die bei Bedarf zugeschaltet wird und den Induktivitätsverlust des Transformators kompensiert. Darüber hinaus kann dadurch die Resonanzfrequenz herabgesetzt werden. Das Problem an diesem Kniff ist die komplizierte Umsetzung, denn ein Mikrocontroller muss hier Frequenz und Periodendauer überwachen und im Bruchteil einer Sekunde aneinander anpassen. So entstehen mehrere der Leistung angepasste Stufen, die durch die Pulsweitenmodulation dann feinjustiert werden können.

Dank der Portierung der LLC-Resonanzwandler-Topologie in moderne Netzteile lassen sich mittlerweile Effizienzwerte von über 90 Prozent erreichen. Titanium- oder Platinum-Spannungswandler wären ohne diese Schaltungstopologie bisher so überhaupt nicht möglich gewesen.

## Wie sich der Wirkungsgrad mit der Auslastung ändert

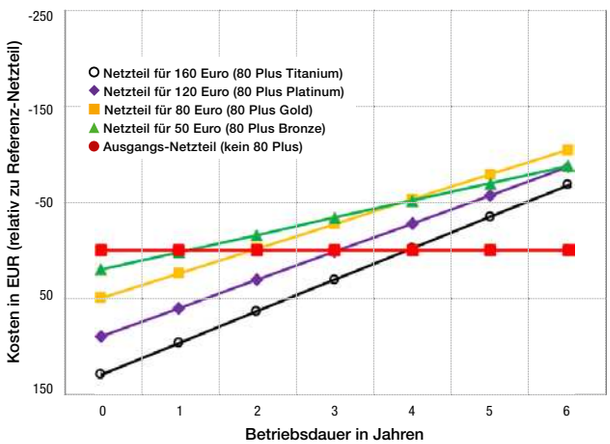
**Der Schwachpunkt aller PC-Schaltnetzteile ist der vergleichsweise schlechte Wirkungsgrad bei niedriger Auslastung – egal in welcher Effizienzklasse. Beachten Sie aber, dass die Y-Achsen-Skalierung erst bei 84 Prozent beginnt.**




## Wann sich ein neues Netzteil amortisiert hat

**Ein älteres Netzteil ohne Zertifikat und nur 80-prozentigem Wirkungsgrad bei halber Auslastung ist in unserem Beispiel bereits vorhanden (rote Linie mit ausgefüllten Punkten) und soll ersetzt werden.**

**Einsparung durch Effizienz (500 W Ausgangsleistung, 250 W feste Last, 6 Std. Betrieb am Tag, 29 Cent/kWh)**




**Die 50 Euro für ein 80-Plus-Bronze-Netzteil haben sich binnen Jahresfrist amortisiert. Wer mehr Geld ausgibt und ein Gold-Netzteil kauft, muss anfangs mehr investieren, übertifft nach vier Jahren aber schon die Ersparnis des Bronze-Gerätes. Das Platinum-Netzteil wiederum holt das Gold-Gerät nach knapp sechs Jahren ein. Einzig das Titanium-Netzteil ist aufgrund seines hohen Preises im Nachteil.**

# So funktioniert die Infrastruktur: Soundkarten

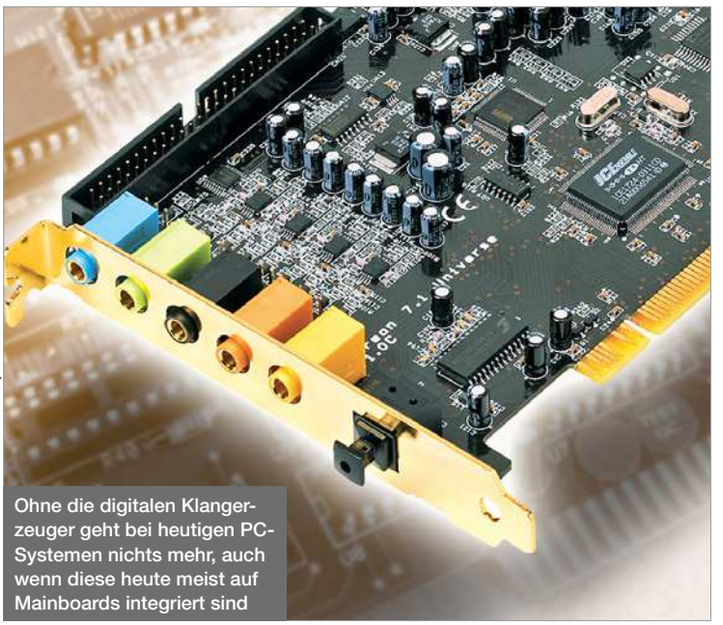


Bild: Merck KGaA, Darmstadt Germany

Ohne die digitalen Klangerzeuger geht bei heutigen PC-Systemen nichts mehr, auch wenn diese heute meist auf Mainboards integriert sind

## Hintergrund: USB-Soundkarten

**Sound-„Karten“ sind seit einigen Jahren auch als USB-Variante zu haben. Dies hat eine Reihe von Vorteilen. Dass man nicht das Gehäuse öffnen muss, ist nur einer davon. In kleinen Gehäusen schützen USB-Soundlösungen vor Überhitzung und die Signalqualität ist etwas besser. Bei internen Soundkarten kann es trotz häufig sehr guter Abschirmung gegen elektromagnetische Impulse immer zu Störgeräuschen kommen; meist werden diese von dem Netzteil oder der Grafikkarte induziert. Schließlich lässt sich USB-Sound auch schnell von einem Gerät zum anderen transportieren; genau das spricht aber auch gegen USB-Sound: Viele Anwender möchten die Sound-Kabel lieber hinter dem PC verstauen als noch ein weiteres Gerät auf dem Schreibtisch unterbringen zu müssen.**

Vor knapp 27 Jahren war der PC noch weit weg vom heutigen Multimedia-Allekönner. Seine Soundambitionen beschränkten sich auf einen kleinen, quäkenden Speaker, der meist im Gehäuse des PCs untergebracht und mit dem Mainboard verbunden war.

### Erste Sound-Schritte

Dennoch brachten es die findigen Spiele-Programmierer auch damals schon fertig, dem PC-Speaker mehr als nur sinusgleiche Beep-Beep-Töne zu entlocken. Wer damals als technikbegeisterter Steppke „Test Drive 1“ auf dem XT gespielt hat, mag sich gefragt haben, wie man das jemals realistischer hinbekommen könnte. Heute wissen wir, dass Test Drive 1 auf dem XT nicht die letzte Ausbaustufe der Evolution war, doch während der große Schritt auf dem 3D-Sektor noch einige Jahre auf sich warten ließ, waren die frühen 90er-Jahre die Geburtsstunde der Soundkarten – genau genommen der PC-Soundkarten, denn auf den Amiga- oder Atari-Rechnern gehörte die Soundkarte seit jeher zur Grundausstattung. Der erste Sound-Blaster-Chip wurde schließlich 1987 veröffentlicht und feierte wegen seines Preises und seiner Aufrüstbarkeit bald große Erfolge.

### Sound Blaster

Der Markenname Sound Blaster ist heute noch der Inbegriff einer Soundkarte. Seinen Markterfolg verdankt die Karte jedoch nicht vorwiegend dem Original, sondern den vielen billigen No-Name-Produkten, die Anfang bis Mitte der 90er-Jahre als Sound-Blaster-kompatible Karten den Markt überschwemmten und in einer Zeit, als es noch keine genormten Multimedia-Schnittstellen gab, einen Quasi-Standard begründeten.

Bis etwa Ende der 90er-Jahre enthielten die Karten oft auch noch einen FM-Synthesizer-Chip, der das Abspielen von elektronischer Musik durch Klangsynthese und ohne die Nutzung von Samples (digitalisierten Klängen) ermöglichte. Der nächste, die Klangqualität verbessernde Schritt war die Implementierung einer einheitlichen Schnittstelle zur Erweiterung der Soundkarte mittels Wavetable. Aufgrund stetig steigender Rechenleistung wurden jedoch sowohl der FM-Synthesizer-Chip als auch die Wavetable-Erweiterungs-Option schließlich unnötig.

### Funktionsweise

Die beiden entscheidenden Bauteile auf einer Soundkarte sind zum einen der digitale Sound-Prozessor DSP und zum anderen die Kombination aus Analog-Digital(ADC)- sowie Digital-Analog-Converter(DAC)-Chip. Das Arbeitstier ist dabei der DSP. Er nimmt dem Prozessor eine Menge Last bei der Bearbeitung von Audiodaten ab, die von der Festplatte oder vom CD-ROM-Laufwerk kommen.

Im DSP werden Algorithmen wie Filter, diskrete Fourier-Transformation oder PID-Regelung eingesetzt. Elementare Operationen, aus denen der Algorithmus zusammengesetzt ist, sind dabei zum Beispiel die gliedweise Addition von Signalwerten, die gliedweise Multiplikation von Signalwerten mit einer Konstanten, die Verzögerung (Zeitverschiebung) eines Signals sowie weitere mathematische Operationen, die periodisch aus einem Ausschnitt eines oder mehrerer Signale

einen neuen Wert und aus diesen Werten ein neues Signal generieren.

## DSP und DAC

Je mehr dieser Funktionen der DSP einer Soundkarte beherrscht, desto weniger wird die CPU damit belastet, was zum Beispiel in aufwendigen 3D-Spielen von Vorteil ist, da somit mehr Rechenzeit für KI und Grafik zur Verfügung steht. Damit die bearbeiteten Audiodaten anschließend auf den Boxen oder der Soundanlage wiedergegeben werden können, müssen sie von binären 0101-Sequenzen in analoge Signale übersetzt werden. Diese Funktion übernimmt der DAC. Soll von einer externen Audioquelle aufgenommen werden, ist dafür der ADC zuständig, der die Signale digitalisiert und an den DSP zur Weiterverarbeitung und anschließenden Speicherung auf der Festplatte schickt.

## Karte vs. Platine

Heute ist die Zeit der teuren Soundkarten auf dem Massenmarkt längst vorüber. Die Soundkarte ist ein Nischenprodukt für Audioprofis und Klangfetischisten geworden. Der Rang wurde ihr bereits vor etlichen Jahren durch den Onboard-Sound abgelaufen. Meist werden kleine und preiswerte Codec-Chips verbaut, welche über ein standardisiertes Interface mit den in der Southbridge des Chipsatzes bzw. dem PCH integrierten Schnittstellen verbunden sind.

In Sachen Codec-Chips hat sich heute die Firma Realtek ein Quasi-Monopol erarbeitet. Auf vielen aktuellen Mainboards sind Chips oder Chip-Designs von Realtek verbaut. Auch in diversen Ablegern wie Avance- oder ADI-Chips steckt Realtek-Technologie.

## AC97 und HD Audio

Damit eine Firma wie Realtek überhaupt einen funktionierenden Codec-Chip auf dem Markt anbieten konnte, bedurfte es zunächst eines Standards. Als AC97 (Kurzform für Audio Codec 97) bezeichnet man Intels Audiostandard, der von den Intel Architecture Labs im Jahr 1997 entwickelt wurde und hauptsächlich von Onboard-Soundkarten, Modems und Soundkarten genutzt wird. Der Standard definiert eine hochqualitative 16- oder 20-Bit-Audio-Architektur für den PC, die in einem großen Teil heutiger Desktop-PCs vorhanden ist. AC97 unterstützt 96 kHz Abtastrate in 20 Bit Stereo und 48 kHz in 20 Bit Stereo für Mehrkanal-Aufnahme und -Wiedergabe. Ab Version 2.3 unterstützt AC97 Plug-and-Play-Audio für den Endnutzer. Diese Version gibt Auskunft über das angeschlossene analoge Gerät (etwa Boxen, Kopfhörer, Mikrofon).

AC97 wurde vor einigen Jahren vom „High Definition Audio Interface“ (HDA) von Intel abgelöst; die Mindestanforderungen an die Sound-Ausgabe wurden erhöht, die Bandbreite beträgt nun bis zu 192 kHz, 32 Bit und Multichannel-Audio verbesserten den Klang weiter. HDA ist noch heute Standard.

# Wie Sound auf dem Mainboard integriert ist

### Anschlüsse 1

Wie bei einer Soundkarte sind mehrere Aus- und Eingänge für Kopförer, Mikrofone und Lautsprecher vorhanden. Gold schützt vor Korrosion, hat aber sonst keine Vorteile.

### Kopfhörerverstärker 2

Wollen Sie Kopfhörer oder ein analoges Headset anschließen, ist ein Kopfhörerverstärker elementar. Dieser hier kann sogar individualisiert werden (gesockelter Op-Amp).

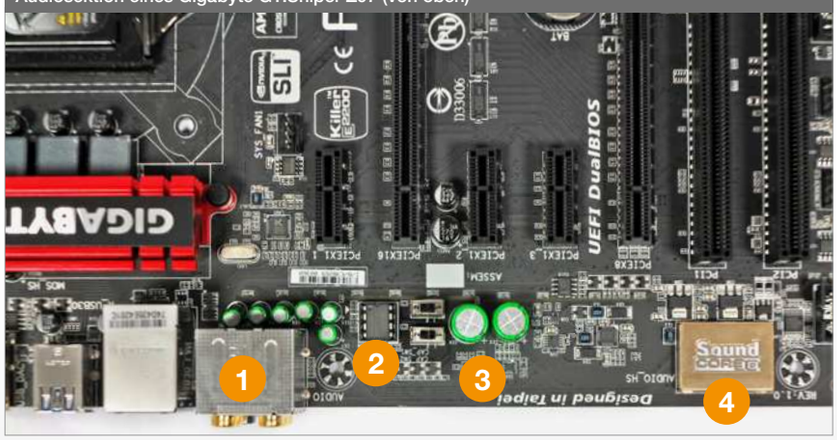
### Komponenten 3

Elektronische Bauteile wie diese Nichicon-Kondensatoren sorgen für eine stabilere Stromversorgung und besseren Klang.

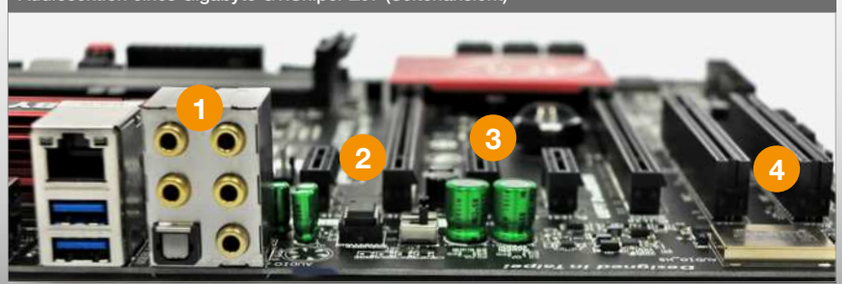
### Soundprozessor 4

Das Herz von Soundkarte und Onboard-Lösung ist der Soundprozessor. Oft ist dies ein Realtek-Codec, hier wurde aber ein Creative-Chip verbaut.

Audiosektion eines Gigabyte G1.Sniper Z97 (von oben)




Audiosektion eines Gigabyte G1.Sniper Z97 (Seitenansicht)



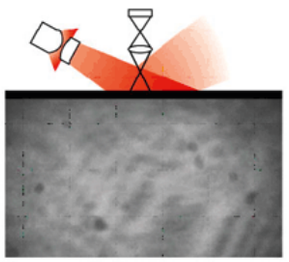
# So funktioniert die Infrastruktur: Mäuse

Bilder: Corsair, Glorious PC Gaming Race

## LED- und Lasersensoren

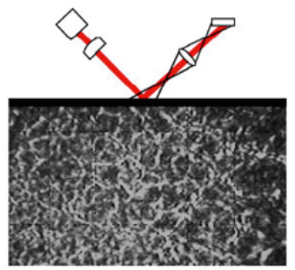
**Mit der LED-Beleuchtung erkennt der Maussensor deutlich weniger Untergrundinformationen als mit der wesentlich feineren Laserbeleuchtung.**

LED-Abtastung



Das LED-Licht tastet weniger Untergrundinformationen ab.

Laser-Abtastung



Die Laser-Beleuchtung kann den Untergrund höher auflösen.

Seit die meisten modernen Betriebssyteme eine grafische Benutzeroberfläche (GUI) besitzen, ist die Maus eine der wichtigsten Mensch-Maschine-Schnittstellen. Was mit der 1968 vorgestellten Ur-Maus begann, entwickelte sich dank immer genauerer Abtasttechniken und der Anpassung an die kontinuierlich modernisierten PC-Schnittstellen (RS232, PS/2, USB) zum hochpräzisen Steuergerät für Spiele und Anwendungen.

### Funktionsprinzip

Das grundlegende Anwendungsprinzip der Computermaus, deren Erfolgsgeschichte 1984 mit der GUI des Apple Macintosh begann, ist über die Jahre hinweg bis heute annähernd gleich geblieben: Mäuse setzen die Bewegungen auf dem Schreibtisch zweidimensional in Veränderungen auf der X- und Y-Achse um. Dabei nimmt ein Sensor im Inneren die Mausbewegungen auf, digitalisiert sie und überträgt diese dann über eine Schnittstelle an den PC. Mithilfe von Betriebssystemfunktionen wird diese zweidimensionale Bewegungsinformation in eine adäquate Bewegung des Mauszeigers auf dem Bildschirm umgesetzt.

### Optomechanische Mäuse

Zwischen 1980 und Ende der 90er-Jahre beherrschte die klassische Kugelmaus den Markt. Das auch als optomechanische Maus bekannte Gerät ermittelt die Bewegung auf dem Tisch durch die Rotation einer Rollkugel im Gehäuse (siehe Bild rechts). Diese überträgt ihre Bewegungen auf zwei schmale Walzen, die seitlich und unten an der Kugel anliegen. Dabei ist eine Rolle für die X-, die andere für die Y-Achse zuständig. Die Achsenrotation wird entweder durch eine Lochscheibe und einen lichtempfindlichen Sensor am Ende der Achse oder durch Kontakte auf der Lochscheibe und den dazugehörigen Impulsnehmer in digitale Signale umgewandelt.

Der größte Nachteil dieser Technik: Kugelmäuse sind sehr anfällig für Verschmutzungen, da die Kugel immer wieder Staub- und Dreckpartikel aufnimmt, die dann an der Mechanik haften bleiben. Des Weiteren führt die Massenträgheit der Kugel dazu, dass bei sehr schnellen Bewegungen oder glatten Untergründen der „Grip“ der Kugel abreißt.

### Bis heute im Einsatz: optische Abtastung

Die wohl wichtigste Weiterentwicklung der Maus-Technik ist der Wechsel zur direkten optischen Untergrundabtastung. Hier wird der durch eine Lichtquelle beleuchtete Bereich unter der Maus von einer Art Mini-Kamera erfasst. Die Richtung und Geschwindigkeit der Bewegungen berechnet dann ein zur Kamera gehöriger digitaler Signalprozessor aus dem Unterschied zwischen nacheinander aufgenommenen, in Graustufen vorliegenden Bildern.

Als Berechnungs-Algorithmus wird „optischer Fluss“ verwendet. Für die Beleuchtung des Untergrunds ist entweder eine LED- oder eine Laserdiode verantwortlich. Dabei bietet die Laser-Beleuchtung dank des sogenannten „Speckle-Effekts“ sowie der sich zeitgleich mit den Lasermäusen etablierenden Kombination aus hochauflösenden Sensoren und Linsen eine feinere Auflösung

des Untergrunds. Im Gegensatz zu ihren LED-Pendants funktionieren Lasermäuse sogar auf sehr glatten Oberflächen ohne erkennbare Struktur problemlos. Sowohl Microsoft als auch Logitech entwickelten beide Techniken weiter. So arbeiten zum Beispiel Logitechs Lasermäuse mit Darkfield-Laser-Tracking-Technologie seit 2009 sogar auf Glas.

## Kommunikation mit dem PC

Mit dem 1987 von IBM entwickelten PS/2-Anschluss, der synchron/seriell arbeitet und bei dem die Informationen bitweise von der Maus an den Rechner übertragen werden, sind aktuelle Geräte nicht mehr ausgestattet. Sie kommunizieren via USB-Anschluss mit dem PC. Bei USB kommen insgesamt vier Kabel (PS/2: eine Datenleitung) zum Einsatz, von denen zwei zur Datenübertragung und zwei zur Versorgung der angeschlossenen Komponente mit Strom dienen. So können USB-Geräte im laufenden Betrieb am PC angeschlossen und entfernt werden.

Auf den miteinander verdrillten Datenkanälen werden ein Signal sowie eine invertierte Variante des Signals gesendet. Aus der Differenz dieser beiden Signale bildet der Empfänger die gesendete Information. Kabellose Mäuse funktionieren in puncto Abtastung wie ihre verdrahteten Brüder. Allerdings muss eine drahtlose Maus eine eigene Stromversorgung sowie einen Sender besitzen, der die Signale via Funk/Infrarot (27-MHz-Band) oder Bluetooth (2,4-GHz-Bereich) an die per PS/2 oder USB mit dem PC verbundenen Basisstation überträgt.

## Dpi- und Polling-Rate

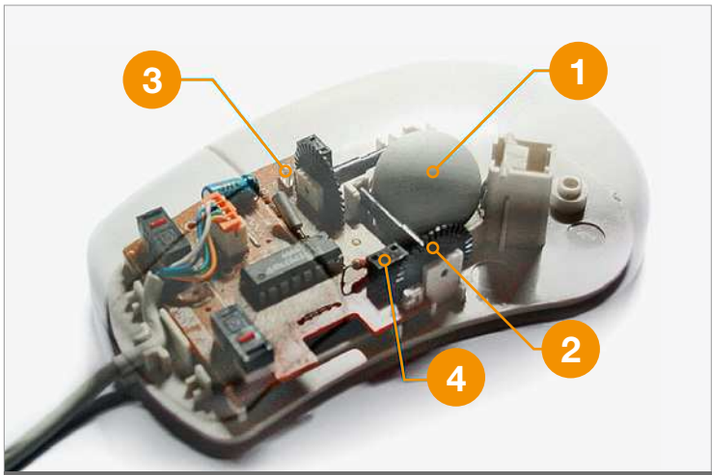
Die am häufigsten verbreitete Maßeinheit für die Leistung des Mayssensors ist Dpi (dots per inch). Der Dpi-Wert gibt an, wie viele Schritte (Pixel) eine Maus innerhalb eines Zolls (= 2,54 cm) auf dem Bildschirm zurücklegt. Zur Verdeutlichung: Eine 1.600-Dpi-Maus bewegt sich pro Zoll 1.600 Pixel auf dem Monitor – ohne Beeinflussung der Mausbewegung und -beschleunigung durch die zwischengeschaltete Software.

Ein weniger verbreiteter, aber technisch korrekter Terminus für die Abtastrate ist Counts per Inch (Cpi). Cpi bestimmt die Zahl der Counts, die der Sensor aufnimmt, wenn das Eingabegerät physikalisch um einen Zoll verschoben wird. Der PC erfasst die Counts als Bewegungsdaten und setzt sie auf dem Desktop des OS sowie bei Spielen mit Maussteuerung als Bewegung einer bestimmten Pixelmenge um.

## Polling Rate

Die Polling-Rate – auch als Signalrate oder Abtastgeschwindigkeit bekannt – gibt im Gegensatz zum Dpi-/Cpi-Wert an, wie viele Abfragen pro Sekunde die Maus bei Bewegungsänderung und Betätigung der Tasten über den USB-Port und den Treiber an den PC sendet. Je häufiger diese Meldungen an den Rechner gehen, desto zügiger kann dieser Bewegungen oder das Auslösen der Tasten weiterverarbeiten.

Eine höhere Polling-Rate ist also tendenziell „besser“, wenn es auf eine sehr schnelle sowie präzise Reaktion der Maus in Spielen ankommt. Standardmäßig ist via USB-2.0-Anschluss lediglich eine Abtastgeschwindigkeit von 125 Hertz möglich. Topmäuse werden dagegen mit sehr leistungsstarken (Laser-)Sensoren ausgestattet sowie mit speziellen Treibern geliefert, die eine Polling-Rate von bis zu 2.000 Hz ermöglichen. Dabei können Nutzer die Einstellung der Signalrate in der Regel in der Software vornehmen.




Bei der Optomechanik gibt die Kugel (1) die Bewegung an die Achse samt Lochscheibe (2) weiter. Durch die LED (3) und den Fotosensor (4) ermittelt die Maus die Richtung.

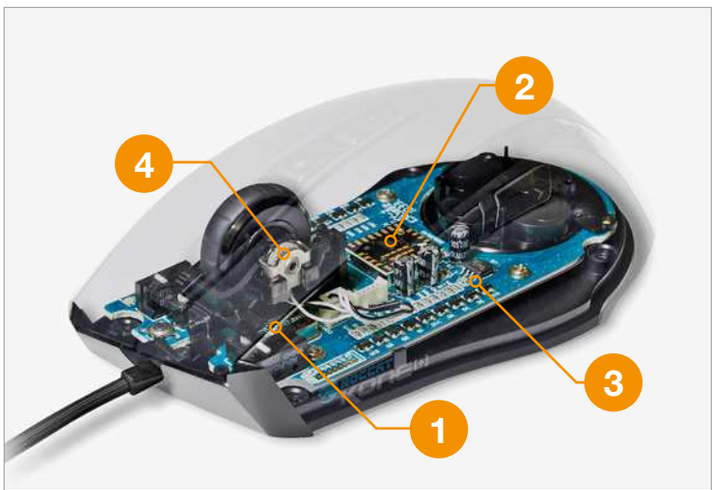
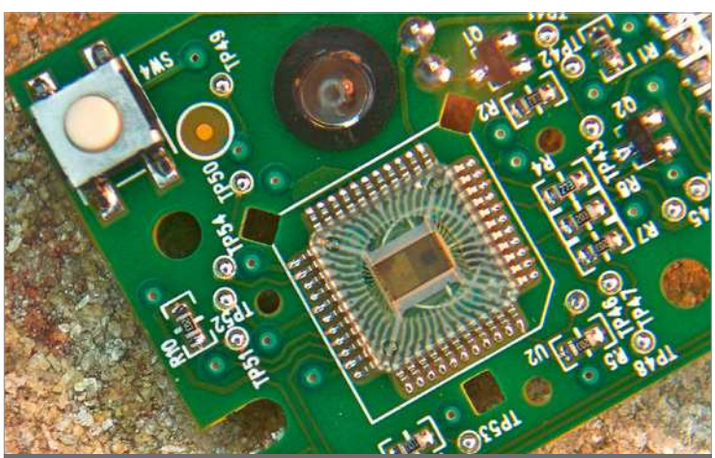



Bild: Roccat

Bei der Lasermaus berechnet der Prozessor (2) mit per Sensor (1) aufgenommenen Bildern Richtung und Bewegung. Ebenfalls im Bild: Profilspeicher (3) und Mausrad-Encoder (4).



In der Mitte des Chips befindet sich die kleine Kamera, die Veränderungen der Oberfläche erkennt und an den Steuerchip weitergibt.

# So funktioniert die Infrastruktur: Tastaturen

Die meisten Tastaturen besitzen eine sogenannte Dome-Switch-Matte oder einzelne Gummidome, deren elektrisch leitende Unterseite auf die Platine gepresst wird und zwei Kontakte überbrückt. Die Rückführung der Taste erfolgt ohne spezielle Mechanik.

Auch wenn immer neue Tastatur-Varianten auf den Markt kommen – das grundlegende Funktionsprinzip des Keyboards hat sich in den letzten 20 Jahren kaum geändert.

## Aufbau & Mechanik

Bei den meisten PC-Tastaturen befindet sich im unteren Teil des Gehäuses eine leitfähig bedruckte Polyesterfolie. Diese ist mit einer Platine verbunden, auf welcher der Steuerchip sowie bei Modellen mit Makro- und Profilverwaltung ein Speicherchip gelötet ist. Auf der Folie mit den Leiterbahnen befindet sich eine Matte mit sogenannten Gummidom-Schaltern – bei einigen Keyboards kleben die Gummidome auch einzeln auf der Spezialfolie. Beim Herunterdrücken der Tasten wird der auf der Unterseite elektrisch leitende Gummidom auf die Polyesterfolie gepresst und überbrückt so zwei Kontakte. Zu diesem Zweck besitzt die Taste auf der Unterseite einen Dorn, der im oberen Gehäuse (Chassis) befestigt ist und durch Führungen an Ort und Stelle gehalten wird – Tasten, die mehr als einen Rasterplatz belegen, besitzen zusätzlich Bügel, die einen Lageausgleich gegen Verkanten unterstützen. Der durch den Tastendruck/das Pressen des Gummidoms auf die Folie entstehende Impuls wird an den Steuerchip weitergegeben, der die Information dann zum PC weiterleitet.

Bei Geräten mit flachen Tasten (zum Beispiel bei vielen Notebook-Tastaturen) werden die Tasten beim Druck auf den Gummidom durch eine Scherenmechanik geführt. Eine in der Herstellung deutlich teurere Alternative zu Tastaturen mit Gummidom-Schaltmechanismus sind mechanische Pendants, deren Tasten bespielsweise einen mit einer Blattfeder versehenen mechanischen Schalter betätigen (Cherry-MX-Schalter) oder eine Feder innerhalb der Tastenführung herunterdrücken, die einknickt und dann einen kleinen Hammer auslöst, der den elektrischen Kontakt auf der Polyesterfolie herstellt (Buckle Springs). Des Weiteren gibt es Geräte mit kapazitiven Kontakten, elektromechanischen Kurzhubtasten, Karbonkontakten mit Kunststoffstößel und Silikonhaube, einer flexiblen Folientastatur mit Silikonüberbau oder Silikontasten.

## Die Luxusklasse

Mechanische Keyboards sind bei Vieltippern, aber auch bei Spielern inzwischen wieder sehr beliebt. Die robusten Geräte haben in den vergangenen Jahren eine Renaissance erlebt. Fans mechanischer Tastaturen lieben deren ganz speziellen Anschlag und Druckpunkt. Anstatt wie bei der Gummidom-Mechanik die Taste vollständig herunterdrücken zu müssen, sind mechanische Tastenschalter so angelegt, dass sie bereits anschlagen, bevor die Taste komplett durchgedrückt wird. Das spart Kraft und beugt Ermüdungserscheinungen vor, da wirklich nur der in der Maßeinheit Gramm angegebene Aktivierungsdruck für das Auslösen des Schalters benötigt wird. Für eine taktile Rückmeldung sorgt eine Blattfeder mit einem Druckpunkt.

## Tastaturtypen

Der bei mechanischen Tastenbrettern am häufigsten zu findende Schaltertyp ist der Cherry-MX-Schalter, den es in verschiedenen Farbkodierungen gibt, die sich bezüglich des Aktivierungsdrucks

und der taktilen Rückmeldung unterscheiden. Beispielsweise besitzt die blaue und von Vielschreibern besonders geschätzte Variante einen Aktivierungsdruck von ca. 50 g und verfügt über eine taktile Rückmeldung inklusive Klickgeräusch. Schwarze, von Spielern bevorzugte „Cherrys“ dagegen arbeiten linear und geben kein akustisches Feedback (Aktivierungsdruck: ca. 60 g). Neben weiteren Cherry-Schaltern in Braun, Transparent und Rot gibt es noch die sogenannte Buckle-Springs-Technik sowie ALPS- und TOPRE-Schalter.

## Matrix-Technologie

Da Keyboards bereits im Standard-MF2-Tastenlayout über 104 Tasten verfügen, ist es wenig sinnvoll und in der Fertigung zu kostspielig, jede Taste mit einem einzelnen Kabel zum Mikrocontroller, dem Steuerchip, zu leiten. Aus diesem Grund setzte sich die Matrix-Technologie durch, bei der die elektrischen Leitungen auf der Polyesterfolie in Reihen und Spalten angeordnet sind – beim MF2-Keyboard mit 104 Tasten sind das zum Beispiel 13 Spalten und acht Reihen. Im Betrieb überprüft ein Mikrocontroller in der Tastatur regelmäßig alle Reihen und Spalten auf Veränderungen im elektrischen Fluss. Durch den Druck einer Taste wird eine bestimmte Zeile mit einer bestimmten Spalte elektrisch verbunden und der Steuerchip kann durch die Beschaffenheit des Signals diesem einen einzelnen Tastendruck zuweisen.

## Problemfall: Ghosting

Je nach Bauart des Tastenbretts und in Abhängigkeit von der Anordnung der Tasten in der Matrix erkennt das Gerät idealerweise alle gleichzeitigen (n-Key-Rollover), im schlechtesten Fall maximal zwei simultane Tastendrücke (2-Key-Rollover). Gerade bei Geräten mit 2-Key-Rollover überschneiden sich die verschiedenen Leitungen der Matrix. Werden hier drei Tasten gleichzeitig gedrückt, registriert der Controller irrtümlicherweise einen vierten Tastendruck, ein Phänomen, das als (Key-)Ghosting bezeichnet wird. Moderne Tastaturen erkennen den Ghosting-Effekt und ignorieren den dritten Tastendruck – man spricht vom Key-Jamming. Für PC-Spieler ist das besonders ärgerlich, wenn sie neben der Bewegung auch die Waffenwahl und Sprünge über Tastenkombis ausführen. Eine Gegenmaßnahme ist es, die Tastenmatrix so anzuordnen, dass es Zonen gibt, in denen die beim Spielen am häufigsten benötigten Tastenkombinationen keine Erkennungsprobleme verursachen, sprich sich die Anordnung der Leitungen für Tasten auf der Matrix nicht kreuzen.

## PS/2 kontra USB

Kommuniziert der Tastatur-Controller per PS/2-Anschluss mit dem Rechner, sendet er sowohl beim Betätigen als auch Loslassen der Taste einen Code, der im PC einen Interrupt auslöst. Die Interrupt-Behandlungsroutine wertet den Scancode der gedrückten Taste aus, ermittelt, ob zusätzlich Modifier-Tasten (Shift/Strg/Alt) aktiv sind, und speichert das Ergebnis anschließend im Tastaturpuffer ab. Mit Software-Interrupts arbeitet das OS diesen Puffer dann ab und verknüpft den Scancode mit einem Zeichen (Keycode).

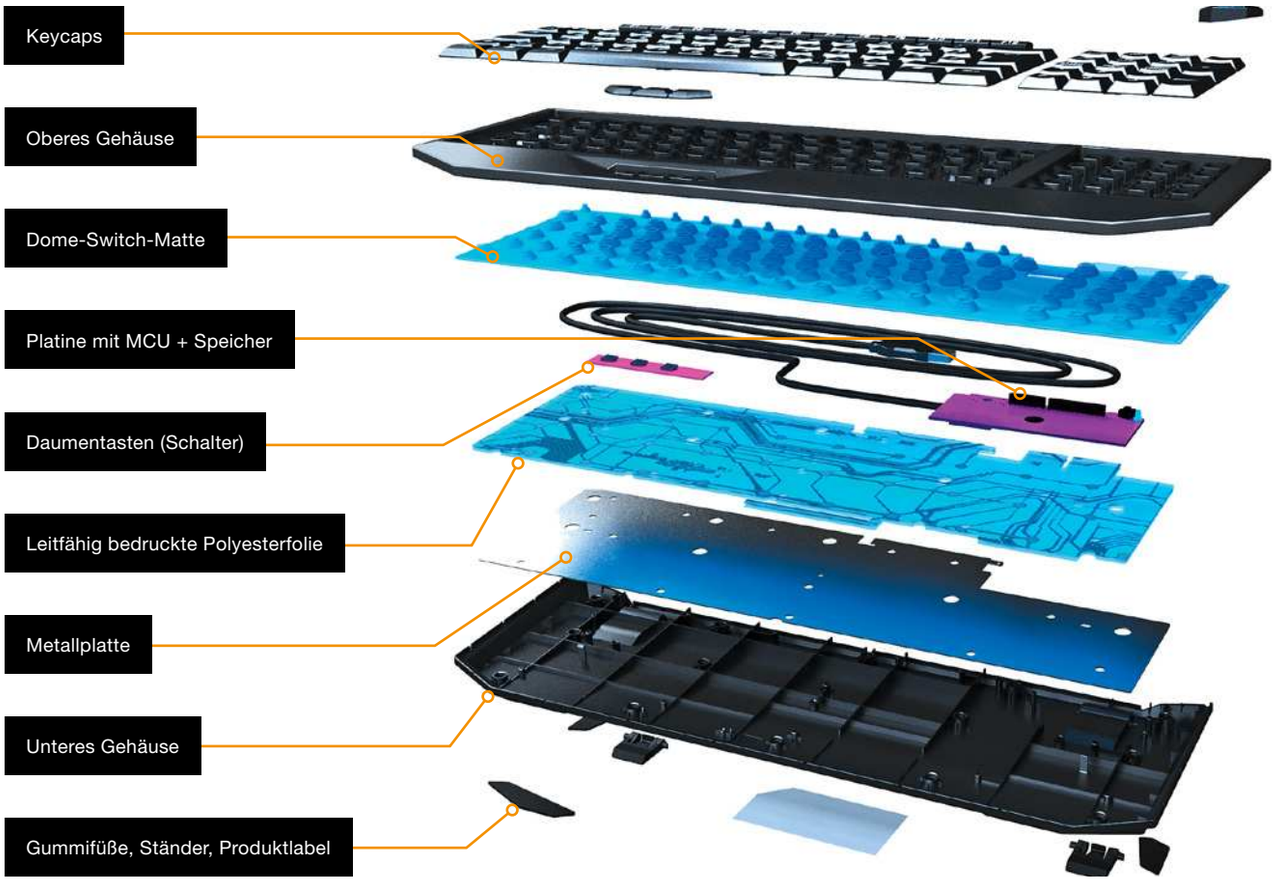
Am USB-Anschluss behandelt der Host-Controller die Tastatur als Human-Device-Interface-Gerät. Anders als Geräte mit PS/2-Anschluss sendet die USB-Tastatur nicht jedes Mal, wenn der Nutzer eine Taste betätigt oder loslässt, sogenannte Make- bzw. Break-Codes. Sie transferiert einen Datenblock mit Statusinformationen (Flags) für die Umschalttasten und Bytes für die sonstigen Tasten, in dem der gesamte aktuelle Tastatur-Status angegeben wird.

Tasten mit Scherenmechanik besitzen eine scherenartige Führung, die beim Niederdrücken wie ein Liegestuhl zusammenklappt. Diese Mechanik besteht aus zwei geschlossenen Bügeln, die an den Enden über Gelenke miteinander verbunden sind.

Mechanische Cherry-MX-Schalter (hier Typ Blau mit Click) verfügen über eine Blattfeder mit einem Druckpunkt, der eine bestimmte Kraft zur Überwindung der signal- oder kontaktlosen Ruhelage erfordert. Dadurch erhält der Nutzer eine taktile Rückmeldung.

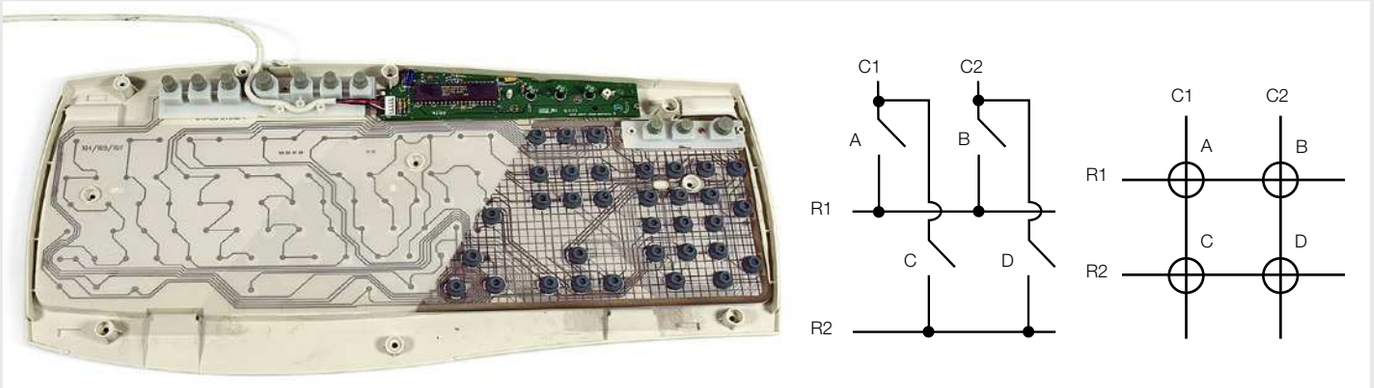
Bei der Buckle-Springs-Technik (IBM Model M) knickt die Feder im Inneren der Tastenführung nach einer gewissen Druckdistanz seitlich ein. Dadurch schlägt ein Hammer an der Unterseite der Führung auf die Membranfolie und stellt so elektrischen Kontakt her.

# Aufbau einer Spieletastatur




## Tastaturmatrix im Detail

**Vor allem beim Spielen kann es vorkommen, dass mehrere Tasten gleichzeitig gedrückt werden. Abhängig von der Anordnung der Tasten in der Matrix (Platine mit in Reihen und Spalten angeordneten elektrischen Leitungen) kommt es dabei zu Fehlerkennungen. Für unser Beispiel verwenden wir eine Vier-Tasten-Tastatur mit zwei Reihen und zwei Spalten. Hier liegen A und B auf der gleichen Reihe (links). Drückt man nun A und C gleichzeitig (rechts), erkennt der Tastatur-Controller die Eingabe B nicht mehr.**



# So funktioniert die Infrastruktur: VR-Brillen

VR-Brillen gehören zweifellos zu den größten Innovationen des PC-Marktes der vergangenen Jahre. Doch der große Durchbruch lässt weiter auf sich warten. Warum?

## Das Konzept hinter VR

Die Idee der VR-Brillen ist bereits sehr alt. Im Kern geht es darum, das Sichtfeld des Nutzers möglichst vollständig mit Bildern einer virtuellen Welt zu füllen. Am einfachsten gelingt das, indem man dem Probanden ein Display möglichst nahe vor das Auge setzt und den restlichen Sichtbereich möglichst gut abdunkelt. Es liegt also nahe, das Ganze in Form einer blickdichten Brille zu realisieren.

## Parallax-Effekt

VR-Brillen versuchen dem Träger vorzugaukeln, er befinde sich wirklich in der dargestellten Welt und unterstützen diese Illusion, indem sie das gezeigte Bild gemäß der Neigung und der Drehung des Kopfes in Echtzeit anpassen. Zudem simuliert die Brille die leicht abweichende Perspektive (Parallaxe), die jedes Auge durch seine zur Mitte des Kopfes versetzten Lage hat, indem sie den Träger der Brille mit einem für jedes Auge leicht angepassten Bild versorgt. Das kann mittels eines geteilten oder zwei getrennten Displays erfolgen. Das Gehirn ist aufgrund dieser Abweichungen in der Perspektive zum räumlichen Sehen in der Lage.

## Das Sichtfeld

Maßgeblich für den gefühlten Realitätsgrad einer VR-Anwendung ist das Sichtfeld. Dieses wird als Winkel angegeben (beispielsweise 105°) und beschreibt, welchen Teil einer Szenerie beide Augen maximal durch die Brille angezeigt bekommen. Vereinfacht gesagt, ist die Darstellung umso realistischer, je besser dieses Sichtfeld der natürlichen Wahrnehmung der Augen angepasst ist (Details siehe Kasten auf der folgenden Doppelseite). Ältere 3D- oder VR-Brillen krankten oft daran, dass das Sichtfeld zu klein und damit auch die Ränder (Das „Ende“ der Brille) klar erkennbar waren.

## Ein neuer Anlauf

Seit einigen Jahren ist vor allem eine VR-Brille in aller Munde, und zwar in erster Linie aus dem oben genannten Grund: Das Sichtfeld wurde so gewählt, dass Ränder kaum noch erkennbar sind, der Grad des „Eintauchens“ in die Spielwelt (Immersion) ist hier besonders hoch. Die „Oculus Rift“-Brille wurde von der Öffentlichkeit sehr positiv aufgenommen, die Entwicklerfirma ein Jahr später von Facebook gekauft.

## Die VR-Technik

Der wichtigste Schwachpunkt der Technik ist die Auflösung. Die wohl prominentesten Modelle, die HTC Vive und die Oculus Rift, nutzen nur ein einzelnes LCD-Panel, wovon je eine Hälfte für ein Auge zuständig ist. Bei der ersten Version von Oculus Rift hatte das Panel eine Auflösung von 1.280 × 800 Pixel, womit pro Auge nur 640 × 800 Pixel zur Verfügung standen. Die Erfahrungen mit diesem Gerät waren zwar eindrucksvoll, die Qualität der Darstellung aber insgesamt bescheiden, da durch die Vergrößerung die Pixelzwischenräume als sogenanntes „Fliegengitter“ und sogar die Sub-Pixel sehr deutlich zu erkennen waren.

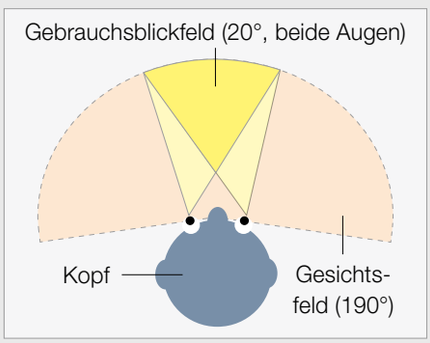
## Ein Tribut an die Vernunft

Aktuelle VR-Brillen lösen mit 2.160 × 1.200 Pixeln auf, was pro Auge

## Kurzes VR-Glossar

**Die wichtigsten Begriffe im Zusammenhang mit VR im Überblick**

### Sichtfeld




**Das Sichtfeld beschreibt den von einem optischen System abgedeckten Raumbereich und wird umgekehrt auch für den Darstellungswinkel von Head Mounted Displays genutzt. Es berücksichtig weder die wechselnde Empfindlichkeit der unterschiedlichen Sichtbereiche des menschlichen Auges, noch dessen Beweglichkeit und auch nicht die Nutzung weiterer Systeme in Form eines zweiten Auges.**

### Gesichtsfeld

**Das Gesichtsfeld vernachlässigt weiterhin die Möglichkeit von Augenbewegungen, bewertet aber die geringere Empfindlichkeit des peripheren Sichtvermögens und optional auch die Nutzung zweier Augen. In letzterem Fall kann das Gesichtsfeld weiter in einen binokularen Bereich mit stereoskopischer 3D-Wahrnehmung und Bereiche, die nur von einem Auge wahrgenommen werden, unterteilt werden. Aufgrund des großen Überlappungsbereiches der Sichtfelder beider Augen beim Menschen liegen diese monokularen Abschnitte aber im peripheren Teil der jeweiligen Sichtfelder und spielen somit für Spieler eine untergeordnete Rolle.**

### Blickfeld

**Blickfelder berücksichtigen zusätzlich Augenbewegungen, wobei auch hier eine Unterscheidung in das Blickfeld jedes einzelnen Auges, das gemeinsame Blickfeld beider Augen und in den darin enthaltenen binokularen Bereich mit stereoskopischem Sehen möglich ist. Hierbei spielen auch nicht-optische Körpermerkmale eine Rolle, so begrenzt zum Beispiel die Nase das Blickfeld der Augen auf die jeweils andere Körperseite. Von besonderer Bedeutung für die VR-Brillen-Entwicklung ist das Gebrauchsblickfeld. Dieser vergleichsweise kleine Bereich ist auch vom individuellen Sehverhalten abhängig und beschreibt den Teil des Blickfelds, der im Alltag zur Fokussierung von Objekten genutzt wird, bevor man an Stelle der Augen den gesamten Kopf schwenkt.**

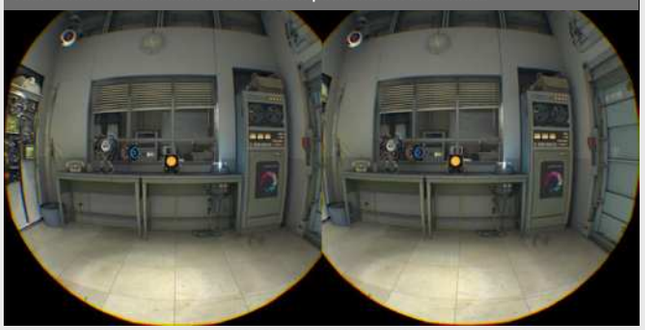
### Umblickfeld

**Wird neben den Augen auch der Kopf bewegt, so spricht man vom Umblickfeld. Dieses ist bei den meisten Menschen nur im unteren Bereich begrenzt, wo der eigene Körper die Sicht einschränkt, während Kopf- und Augenbewegungen in der horizontalen zumindest eine periphere 360°-Erfassung der Umgebung erlauben.**

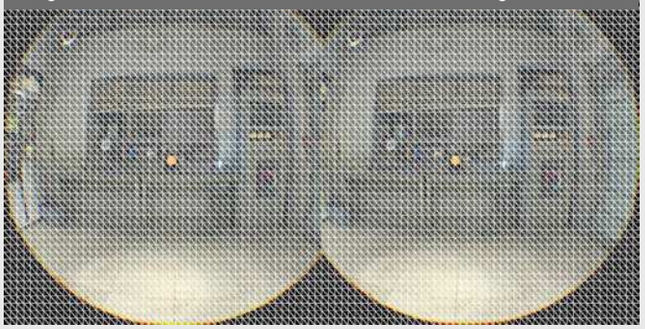
# Wie VR-Brillen Rechenleistung sparen

**Die Verzerrung des Sichtfeldes bei VR bietet eine gute Möglichkeiten zum Einsparen von Rechenleistung: Durch die Kugelform sind Teile des rechteckigen Bildes gar nicht erst sichtbar und können so bei der Darstellung ausgespart werden. Valve gibt einen Einblick, wie das genau funktioniert**

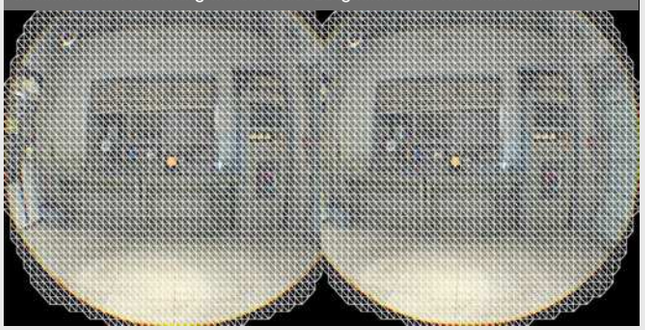
Hier sehen Sie ein einen für VR-Brillen passenden Ausschnitt.



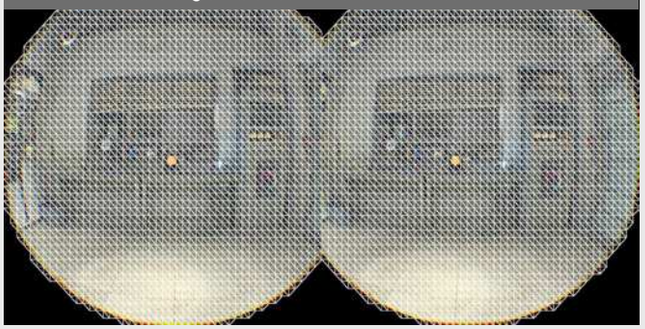
Der gesamte Bildbereich wird mit einem Dreiecksraster überzogen.



In einer ersten Näherung werden erste zu ignorierende Teile entfernt.



Durch weitere Näherung entfallen so 15 Prozent der zu berechnenden Fläche.



effektiv 1.080 × 1.200 Bildpunkte ergibt. Doch auch dies reicht noch nicht für eine perfekte Darstellung. Ein Fliegengitter ist immer noch sichtbar, auch wenn es inzwischen deutlich feiner geworden ist .

Jemand, der nicht so tief in der VR-Materie steckt, fragt sich vielleicht, warum man dann nicht einfach Panels mit höherer Auflösung verbaut. Die Hersteller haben sich hier schlicht vom Pragmatismus leiten lassen, denn die niedrige effektive Bildpunktanzahl täuscht über den notwendigen Rechenaufwand hinweg. Auch wenn die effektive Auflösung beider Brillen niedrig wirkt, entspricht die echte Auflösung eher der eines WQHD- als der eines Full-HD-Display.

Ein Spiel in WQHD kann von modernen Oberklasse-Grafikkarten in den meisten Fällen noch schnell genug dargestellt werden, jedoch sind bei VR höhere Anforderungen zu erfüllen. So muss die Verzögerung zwischen der Kopfbewegung möglichst schnell umgesetzt werden, um Übelkeit beim Anwender zu verhindern.

## Übelkeit und Schwindelgefühle

Viele Menschen neigen nämlich zu Schwindelgefühlen oder Übelkeit, wenn das Gesehene nicht mit den vom Gleichgewichtssinn an das Gehirn übermittelten Wahrnehmungen übereinstimmt. Dieses Leiden kennt man gemeinhin als Bewegungskrankeit (Kinetose). Als Auslöser reichen oftmals bereits Zehntel- bis Millisekunden Abstand zwischen Kopfdrehung und Reaktion des Bildes. VR-Brillen lösen dieses Problem zumindest teilweise, indem sie schnelle Panels mit 90 Hz statt der für LC-Displays üblichen 60 Hz einsetzen. Die 1,5-fache Bildwiederholrate fordert deswegen auch eine entsprechende Steigerung der Rechenleistung, um bei den berechneten Frames pro Sekunden mithalten zu können.

Des Weiteren entspricht die native Auflösung der VR-Brillen nicht der internen Renderauflösung: Die Linsen stauchen das Bild an bestimmten Stellen, weshalb der Ausschnitt etwas höher als die native Auflösung gerendert werden muss. Sowohl Valve als auch Oculus empfehlen einen Faktor von 1,4, was in einer finalen Renderauflösung von 3.024 × 1.680 Pixel entspricht – womit die Brillen schon deutlich über dem Rechenaufwand von WQHD liegen.

## Kantenglättung

Wegen dem großen Winkelmaß der sichtbaren Pixel sowie durch die ständige Bewegung des Kopfes ist Aliasing ein ständiger und auffälliger Begleiter der VR-Brillen. Valve sieht deshalb MSAA 4× als Mindestqualität vor, empfiehlt nach Möglichkeit aber MSAA 8×. Diese Zahlen verdeutlichen, dass die notwendige Rechenleistung nicht weit unter jener liegt, die für UHD bei 60 Fps notwendig ist.

## Alternative Eye Tracking?

VR-Brillen sind mit Preisen bis zu 600 Euro (noch) recht teuer, der Zwischenweg des Eye-Tracking ist wesentlich günstiger zu haben. Eye-Tracking ist eine Technik, bei der nicht die komplette Kopfbewegung in das Spiel übertragen wird, sondern die Blickrichtung der Augen als zusätzliche Steuerungshilfe für den PC dient. Der Anwender eines solchen Systems steuert die Aktionen also damit, indem er Dinge auf dem Bildschirm ansieht. Die Kernkomponente ist hierbei ein stabförmiges Kameramodul, das mit insgesamt drei Infrarot-LEDs ausgestattet ist. Dieses muss an der Bildschirm-Unterseite magnetisch an einem selbstklebenden Metallstreifen befestigt werden.

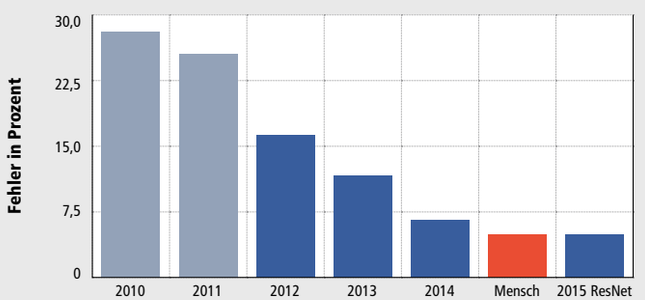
# Software

# So funktioniert Software: Künstliche Intelligenz

Bild: Pixabay

## Überblick: KI-Erfolg

**Seit 2010 gibt es den „ImageNet"-Wettbewerb, bei dem künstliche Intelligenzen ein vorgegebenes Set von Bildern möglichst korrekt erkennen sollen. Innerhalb von fünf Jahren konnten bessere Ergebnisse erzielt werden, als bei der menschlichen Kontrollgruppe. 2012 nutzte AlexNet ein Künstliches neurales Netz (genauer gesagt: ein Convolutional Neural Network, siehe folgende Doppelseite) und markierte damit einen Durchbruch**




Wenn es um mathematische Gleichungen geht, sind Computer in der Lage, diese im Bruchteil einer Sekunde mit Leichtigkeit zu lösen. An vermeintlich einfachen Aufgaben scheitern sie aber, dem intuitiven Erfassen und Erkennen der Umwelt beispielsweise. Doch das könnte sich in Zukunft ändern.

### Intuitive Wahrnehmung

Wie erklärt man einem Computer intuitiv ablaufende Prozesse der Wahrnehmung? Mathematisch ist das Identifizieren von Gesichtern, Objekten, Sprache oder Text nur äußerst schwer beschreibbar, auch weil Menschen dies reflexartig erledigen, ohne einen Gedanken daran zu verschwenden.

Begründet liegt dies in sieben Millionen Jahren Evolution: Je schneller Freund und Feind erkannt werden konnten, desto größer waren die Überlebenschancen und damit der Fortbestand der Spezies. Das ist übrigens auch der Grund für das Phänomen der Pareidolie, das jeder schon mal beobachten durfte: Hier erkennt man Gesichter in Gegenständen, wo eigentlich gar keine sind – nur „passend" angeordnete Strukturen.

### Deep Learning

Eben weil es arithmetisch so schwer zu beschreiben ist, versuchen Wissenschaftler mit sogannten Deep-Learning-Algorithmen gerade, künstliche Intelligenz dazu zu bringen, ähnlich wie Menschen sensorische Rohinformation zu verarbeiten. Erzielt wird das, indem man zunächst die Welt als ein Hierarchiesystem aus Konzepten versteht, in dem sich selbst die komplexesten Szenarien stets aus Blöcken simpler Konzepte aufbauen lassen. Programme sollen damit in Zukunft Wissen aus Erfahrung generieren, indem unbekannte komplexe Konzepte soweit zerlegt werden, bis sie für einen Computer fassbar werden und dann verstanden werden können.

### Selbstoptimierung

Damit werden Computerprogramme bald in der Lage sein, sich selbst zu optimieren. Viel wichtiger ist aber, dass sie bei Datensätzen auf einen menschlichen Interpreten verzichten können, der jede Information für den Computer spezifiziern müsste. Im Zuge des rasanten Wachstums des Internets der Dinge und der damit einhergehenden massenhaften Erzeugung von sensorischen Rohdaten durch die digitale Erfassung der physischen Welt scheint dieser Lernprozess unabdingbar, um der Datenflut überhaupt Herr zu werden.

Überträgt man die einzelnen Beziehungen der Konzepte zueinander in ein Diagramm, bekommt man eine tiefgehende und vielschichtige Visualisierung geboten. In Zukunft werden wir also noch mehr aus diesem Fachbereich der Computerwissenschaften hören, weshalb wir Ihnen auf den folgenden Seiten einen kleinen Überblick geben wollen.

### Maschinelles Lernen

Zunächst sollte geklärt werden, was man überhaupt unter maschinellem Lernen (ML) versteht, da Deep Learning als Teilgebiet des ML sich viele Prinzipien und Techniken zunutze macht.

Den Begriff des maschinellen Lernens prägte Arthur Samuel, ein Pionier auf dem Gebiet der künstlichen

Intelligenz, bereits 1959. Damals führte er Studien zur Mustererkennung in Datensätzen mit Computerprogrammen durch. Computer gewannen bei seinen Experimenten damals aus einem Pool an Beispielen Wissen, da sie innerhalb der Daten Muster und Gesetzmäßigkeiten korrekt erkannten und diese dann auf unbekannte Fragestellungen anwenden konnten.

Man kann also davon sprechen, dass maschinelles Lernen Computer dazu befähigt, aus Erfahrung Wissen zu generieren, und dadurch eine datengestützte Transferleistung erbracht werden kann. Damit lassen sich unbekannte Probleme lösen – wohlgemerkt, ohne vorher spezielle Instruktionen bekommen zu haben.

## Künstliche neuronale Netze

Deep Learning macht sich ähnliche statistische Techniken zunutze wie das maschinelle Lernen. So kann auf ähnliche Weise Wissen aus einen Datenset an Beispielen gewonnen werden. Deep-Learning-Algorithmen versuchen aber, die Welt als ein hierarchisches System aus Konzepten zu verstehen. Dadurch sind deutlich mehr Datensätze nötig, weil auch ein größeres Kontingent an Einzelheiten verstanden werden muss, um beispielsweise einen Hund von einem Wolf oder Fuchs zu unterscheiden.

DL-Algorithmen werden dadurch nicht nur genauer als „reine" ML-Programme, sondern erreichen in der Bilderkennung mittlerweile ähnlich gute oder sogar bessere Resultate als ein Mensch. Erreicht wird das über künstliche neuronale Netze (KNN), die Strukturen im menschlichen Gehirn nachahmen. In die Computerwissenschaften eingeführt wurde der Begriff „Deep Learning" übrigens schon vor mehr als 30 Jahren, von der Forscherin Rina Dechter. Wieso erleben wir also gerade jetzt so einen enormen Hype um das Thema?

## Big Data

Um Deep Learning effizient zu betreiben, benötigt man natürlich zunächst einmal unheimlich große Datenmengen, die von einem Algorithmus untersucht werden können. Diese riesigen Datensätze sind besser bekannt als Big Data, weil sie keine Auswertung mit herkömmlichen Mitteln der Datenverarbeitung zulassen und außer groß auch zu komplex oder zu schnelllebig sind.

Seit Kurzem kann diese Datenflut überhaupt erst generiert und vor allem gespeichert werden. Die unaufhörliche Weiterentwicklung größerer Speicherkapazitäten legte um das Jahr 2002 herum den Grundstein für Big Data, als erstmals mehr Information digital abgespeichert wurde als analog.

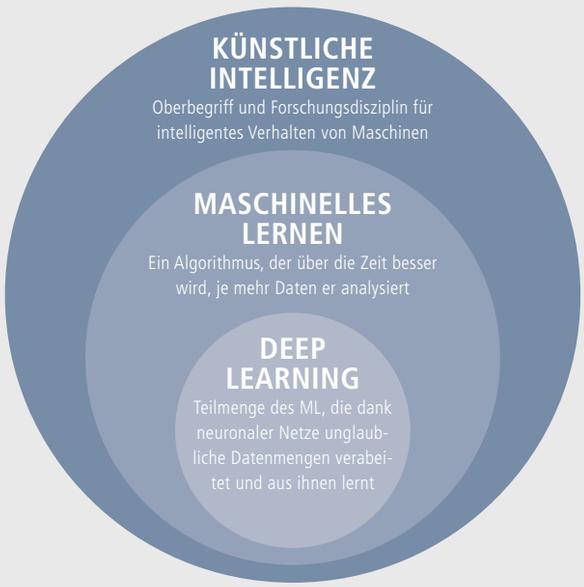
Mit einem vergößerten Angebot an Speicher geht die Informationsexplosion einher: 1986 wurden nur 2,6 Exabytes digital abgespeichert, zur Jahrtausendwende waren es 54,5 Exabytes und 2016 bereits 16,1 Zetabytes (Giga-, Peta-, Exa-, Zeta-). Die IDC (International Data Corporation) und der Speicherhersteller Seagate gehen 2020 sogar von 163 Zetabytes digitaler Informationen aus, von denen gerademal 5,2 Zetabytes durch künstliche Intelligenz analysiert werden – verglichen mit heute ist das ein Zuwachs um den Faktor 50.

## CPU- und GPU-Performance

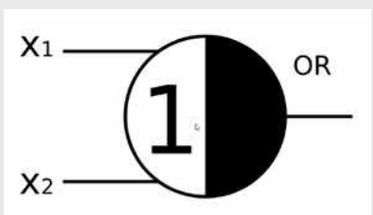
Deep Learning startet aber nicht nur wegen der Big-Data-Datensätze gerade richtig durch, sondern weil Prozessoren und GPUs mittlerweile einen Level an Performance er-

## Überblick: Deep Learning als Teil der KI-Forschung

**Das Forschungsgebiet der künstlichen Intelligenz besteht aus vielen Teildisziplinen. Eine davon ist das Maschinelle Lernen, das wiederum Deep Learning als Untermenge enthält. Andere KI-Disziplinen sind beispielsweise die Robotik.**




## Rückblick: 75 Jahre KI-Forschung




**Angefangen hat die Forschung an Künstlicher Intelligenz bereits in den 1940er-Jahren mit der McCulloch-Pitts-Zelle, dem einfachsten Neuronenmodell überhaupt, welches auch heute noch als sehr effiziente elektrische Schaltung Verwendung findet. Eine der Varianten zeigt die Abbildung oben.**

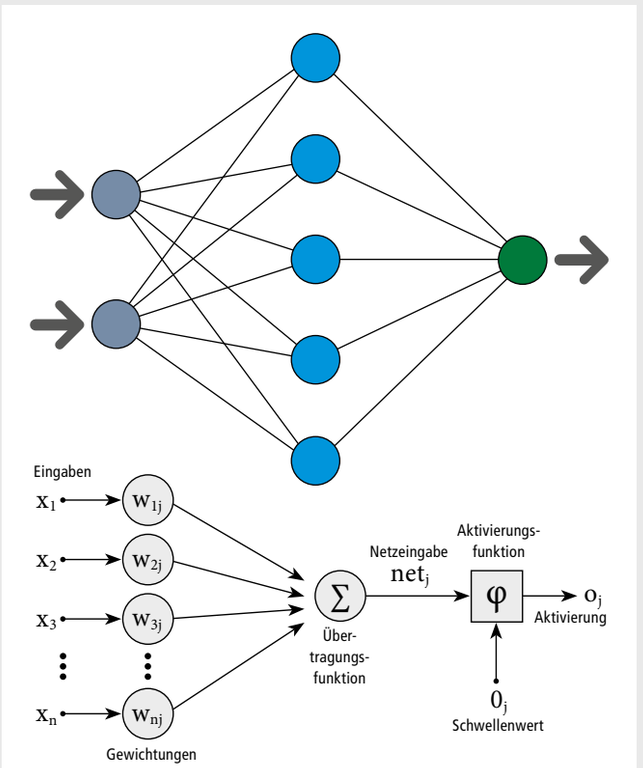
**Wenig später, in den 1950er-Jahren, zeigte Frank Rosenblatt mit dem Konvergenztheorem über das Perzeptron (vom englischen Perception = Wahrnehmung), dass das künstliche Neuron alles erlernen konnte, was es auch abbilden kann, und legte damit den Grundstein für künstliche neuronale Netze.**

**Es blieb Jahrzehnte ziemlich ruhig um KNNs, bis es Geoffrey Hinton 2006 erstmals gelang, ein mehrschichtiges neuronales Netz zu trainieren – seitdem ist das Gebiet wieder in den Fokus der Forschung gerückt. Diese und weitere „Ruhephasen" werden bisweilen auch als „KI-Winter" bezeichnet; in diesen Zeiten gab es teils aufgrund technischer, teils aber auch gesellschaftlicher oder politischer Einschränkungen praktisch keine Fortschritte in der KI-Forschung.**

Bild: Wikipedia/Adrian Lange, Lizenz: CC BY-SA 3.0

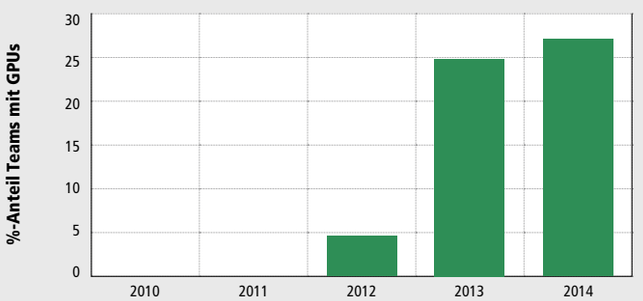
## Überblick: Die Ebenen eines neuronalen Netzes

**Neuronale Netze sind aus drei Ebenen aufgebaut: Input-, Hidden- und Output-Layer. Bei künstlichen Neuronen handelt es sich um mathematische Funktionen, die auf Grund eines Inputs einen Wert errechnen. Mehrere dieser Werte können, analog zur Biologie, einen Schwellenwert in höheren Schichten erreichen und das Neuron dort aktivieren, das dann zum Beispiel für die Identifikation von Personen nötig ist**




## GPUs als KI-Treiber

**Grafikkarten eignen sich aufgrund ihrer massiv parallelen Architektur sowie der stetig steigenden Rechenleistung besonders gut für den Einsatz in neuronalen Netzen. Die GPU-Nutzung beim ImageNet-Wettbewerb illustriert die folgende Statistik:**




reicht haben, der die Verarbeitung gigantischer Datenmengen überhaupt erst erlaubt – vorher war die technische Verarbeitung einfach nicht machbar.

### Wie lernen Computer?

Wir wissen nun, warum gerade jetzt Deep Learning so populär geworden ist und sich zunehmend als Synonym für maschinelles Lernen und künstliche Intelligenz etabliert hat. Noch ist aber unklar, wie genau ein DL-Algorithmus Entscheidungen fällt; beispielsweise wenn Gesichter, Sprache oder handschriftlicher Text erkannt werden soll.

Das ist umso wichtiger, da Computer in naher Zukunft eigenständig und vor allem intelligent handeln können sollen: Ein autonom fahrendes Fahrzeug muss im Straßenverkehr die korrekten Entscheidungen treffen, sonst gefährdet es womöglich andere Verkehrsteilnehmer. Dafür müssen Maschinen lernen, sensorische Rohdaten auf ähnliche Weise aufzubereiten wie Menschen, um Assoziationen knüpfen zu können. Wie soll man aber solch eine Aufgabe bewältigen, wenn die Rohdaten letztlich nur aus Bildpunkten mit unterschiedlichen Farbwerten bestehen?

### Lernbibliotheken

Unabdingbar für den Lernerfolg ist eine gigantische Menge Rohdaten, die als Lernbibliothek fungiert. Dieses Sammelsurium an Beispielen alleine befähigt den Computer aber noch nicht, korrekte Aussagen über ähnliche Sachverhalte zu erstellen. Zunächst müssen Assoziationen zwischen Mustern und Gesetzmäßigkeiten innerhalb der Datensätze erzeugt werden.

Die praktische Umsetzung erfolgt durch Algorithmen, die entweder mit Lehrern (überwachtes Lernen) oder ohne (unüberwachtes Lernen) alle Beispiel-Datensätze durchgehen. Bei der ersten der beiden Möglichkeiten bekommt das Programm stets die korrekten Antworten der Fragestellung mitgeliefert, sodass der Computer leichter Zusammenhänge zwischen Ein- und Ausgabe bilden kann. Das überwachte Lernen hat zudem noch weitere Unterkategorien, bei denen Lernerfolge zum Beispiel belohnt und Fehltritte bestraft werden („bestärkendes Lernen“ kommt am häufigsten beim Mensch vor).

Beim unüberwachten Lernen ist das System dagegen vollkommen auf sich selbst gestellt und muss Strukturen und Muster im Input selbst erkennen. Das kann beispielsweise durch Anhäufung von ähnlichen Merkmalen passieren (Cluster-Analyse). Als Zwischenform gibt es noch das teilüberwachte Lernen, bei dem lediglich ein gewisser Prozentsatz der Antworten zu den Eingaben präsentiert wird.

### Wissen aus Daten

Diese Methoden erlauben es, Computerprogrammen Wissen aus Datensätzen anzutrainieren. Warum glaubt aber ein Algorithmus, auf die korrekte Anwort gekommen zu sein? Wie stellt die Maschine konkret Assoziationen zwischen Ein- und Ausgabe her?

Wie die Entscheidungfindung genau abläuft, ist nicht sicher, da es sich bei DL-Algorithmen um subsymbolische Systeme handelt. Das bedeutet, es kommt zu einer unbewussten Informationsverarbeitung – ähnlich wie beim Denken, da alle psychischen Prozesse in ihrem Wesen unbewusst sind.

Beim klassischen maschinellen Lernen werden dagegen Beispiele und induzierte Regeln explizit repräsentiert, sodass Lösungswege einsehbar sind.

## Entscheidungs-findungen

Damit Computer die Welt ähnlich wie wir erfassen können, haben Wissenschaftler die neuronalen Strukturen des menschlichen Gehirns, genauer die des visuellen Cortex, für eine Maschine nachgebildet.

Bei diesen künstlichen neuronalen Netzen handelt es sich aber nicht um physisch existierende Strukturen, sondern nur um Code, der mathematische Gleichungen ausführt. Meist wird ein Eingabewert in einen Ausgabewert gewandelt und ihm wird dabei eine entsprechende Gewichtung zugeteilt.

## KNN-Aufbau

Künstliche neuronale Netze (KNN) bestehen aus drei abgegrenzten Ebenen: Input-, Hidden- und Output-Layer. Die Eingabeebene registriert zunächst die sensorischen Rohdaten, transformiert diese Information und gibt sie gewichtet an die zweite Ebene, den Hidden-Layer, weiter. Solch eine Gewichtung im Input-Layer kann beispielsweise einen Farbwert eines Pixels widerspiegeln, aber auch das Fehlen von Farbe in einem ganzen Bildbereich.

In der versteckten Ebene passiert dann die „Magie“ eines Deep-Learning-Algorithmus: Hier findet der eigentliche „Denkprozess“ des Computers statt. Von Schicht zu Schicht nimmt die Komplexität und der Detailgrad zu, was erlernt wird. Sind es anfangs noch unterschiedliche Schwarzwerte, werden daraus später Merkmale wie etwa ein Nasenhöcker oder ein Auge und schließlich komplexe Strukturen wie ganze Gesichter.

Bei unüberwachtem Lernen sucht sich der Code eigens die Merkmale in den Bildern aus, bei überwachtem Lernen werden die Gewichtungen innerhalb der Ebenen und Schichten an die durch den „Lehrer” mitgelieferten Ergebnisse angepasst, bis das Resultat den Ausgaben aus dem Lernprozess entspricht. Die einzelnen Schichten und Ebenen sind untereinander verbunden, sodass der Output der n-ten versteckten Schicht der Input des n+1-ten nächsten Hidden-Layers ist.

## Convolutional Neural Networks (CNN)

Eine sehr populäre Herangehensweise für die Bild- und Videoerkennung sind Convolutional Neural Networks (CNN, CovNet, faltende neuronale Netze), die ebenfalls aus Input-, Hidden- und Output-Layer aufgebaut sind. Der Hidden-Layer wird hier aber zusätzlich noch in eine Merkmalerkennungs- und eine Klassifizierungsschicht unterteilt.

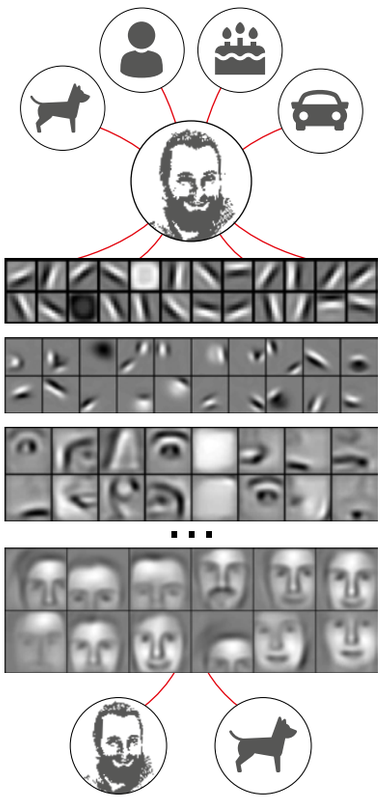
Vereinfacht gesagt geht es darum, einzelne Merkmale der Input-Daten zu „schärfen“. Statt alle Neuronen mit allen zu vernetzen, werden gewissermaßen lokale Untergruppen gebildet. Dies ist vor allem bei der Erkennung von Bildern und Sprache sinnvoll und entspricht im Wesentlichen den Vorgängen im biologischen Vorbild – auch die Zellen der Sehrinde sind verstärkt mit ihren Nachbarn vernetzt. Hinzu kommen noch weitere biologische Merkmale, etwa das „Feuern“ eines Neurons, wenn eine bestimmte Reizschwelle erreicht ist.

## Weitere Entwicklung

Solange die technische Weiterentwicklung anhält, werden auch neuronale Netze zunehmend leistungsfähiger. Wann eine annähernd menschliche Intuition erreicht sein wird, darüber sind Wissenschaftler uneins. Von einigen Jahren bis zu mehreren Jahrzehnten reichen die Prognosen.

# Wie Deep-Learning-Algorithmen Inhalte erkennen

**Mensch, Auto oder doch Hund? Wie bildet ein Computer Assoziationen und lernt, zwischen Mensch, Tier und Objekt zu differenzieren?**




**Übung macht den Meister**

Während der Lernphase werden dem Netzwerk riesige Datensätze mit gelabelten Bildern serviert. Der Algorithmus lernt dadurch, auf bestimmte Merkmale zu achten und Assoziationen zu bilden.

**Der Input**

Dem Netzwerk wird ein unbekanntes und ungelabeltes Bild gegeben, das eingeordnet werden soll.

**Die Hidden-Layer**

In der ersten Schicht des neuronalen Netzes werden zunächst nur simple Strukturen wie Kanten oder Ecken erkannt. Im zweiten Layer wird es dagegen schon etwas konkreter: Die Orientierung der Ecken und Kanten kommt hinzu. In der dritten Schicht kann das Programm dank der vorhergehenden Layer bereits Gesichtsakren ausmachen. Etliche weitere Hidden-Layer später sieht der Computer im Input-Bild ein menschliches Gesicht (Lee et al. 2011; doi :10.1145/2001269.2001295 )

**Präsentation**

Am Schluss gibt der Computer eine Wahrscheinlichkeitsverteilung aus, in der sich die Annahme durch den Algorithmus widerspiegelt.

# So funktioniert Software: Kryptowährungen

Bild: pixabay.com

Als erste Kryptowährung überhaupt hat der Bitcoin im Jahr 2017 die Neunzehntausend-Dollar-Marke durchbrochen – auch wenn der Kurs danach deutlich einbrach, stieg und steigt er doch in der Folge weiter und ist auf dem Weg zum nächsten Allzeit-Hoch. Was steckt hinter dem Boom?

## Kryptowährungen

Viele Leute nutzen mittlerweile Kryptowährungen als Geldanlage oder Zahlungsmittel und können damit sogar schon bei einem Online-Pizza-Portal ihr Essen bezahlen. Aber was steckt genau hinter Bitcoin, Etherum, Litecoin und wie sie alle heißen? Was ist die Blockchain und warum ist sie so sicher? Wir wollen zum Abschluss dieses Bookazines einen kleinen Einblick in die Welt der kryptografischen Währungen geben und Begriffe erklären, damit Sie vielleicht in Zukunft selbst Teil dieser finanziellen Revolution werden können.

## Block für Block Sicherheit?

Die technische Basis jeder Kryptowährung bildet die sogenannte Blockchain. Diese Kette kann kontinuierlich um einzelne Blöcke, die jeweils einen definierten Datensatz enthalten, erweitert werden. Man kann also auch von einer Datenbank sprechen, einem Kassenbuch oder einfach einer Liste, die Transaktionen dokumentiert. Jeder neu erstellte Block wird durch ein kryptografisches Verfahren mit dem vorangegangenen verkettet, um dann unwiderruflich in die Datenbank eingetragen zu werden. Im Gegensatz zu normalen Kassenbüchern bei beispielsweise einer Bank ist die Blockchain dezentral organisiert und kann von jedem überall eingesehen und nachvollzogen werden. Was aber noch viel wichtiger ist: Keine für gültig befundene Transaktion kann wieder gelöscht werden.

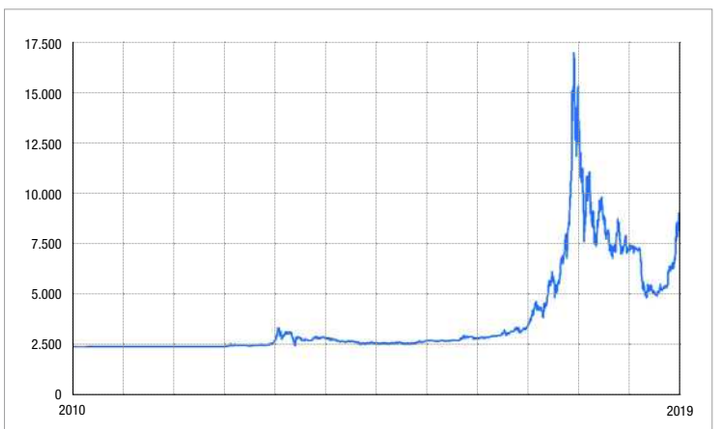
## Blockchain

Fassen wir also noch einmal zusammen: Eine Blockchain stellt eine dezentrale Datenbank dar, die Transaktionen verwaltet, ohne dabei auf eine zentrale Kontrollinstanz zu vertrauen, und das bei vollkommener Transparenz der Eintragungen für jeden Nutzer der Blockchain.

Kommt es nun zu einer Transaktion, wird ein neuer Block in der Kette erzeugt, oder um bei der Analogie mit dem Kassenbuch zu bleiben, wird ein neuer Eintrag hinzugefügt. Dieser neu erstellte Block besitzt alle nötigen Informationen bezüglich der neuen Transaktion, aber auch der gesamten bis dato existierenden Blockchain und damit aller bisher getätigten Überweisungen der Kryptowährung. Nur wenn alle Informationen im neu erstellten Eintrag mit den bisherigen Notizen im Kassenbuch übereinstimmen, ist die Transaktion gültig.

## Dezentrale Kontrolle

Bei der Blockchain kommt anders als in einer Bank eine dezentrale Kontrollinstanz zum Tragen: Es gibt unzählige Kopien der Datenbank, die alle für jede Transaktion einen Prüfprozess durchlaufen müssen. Dafür werden die bisherigen Datensätze mit dem neuen Eintrag abgeglichen und nur bei haargenauer Übereinstimmung wird die Transaktion für gültig erklärt. Stellen Sie sich das ungefähr so vor: Person A möchte Person B Geld senden, kennt Person B aber nicht persönlich und hat deshalb – natürlicherweise – wenig bis kein Vertrauen




Der Bitcoin-Kurs stieg Ende 2017 wie eine Rakete – stürzte aber auch schnell wieder ab. Aktuell steht der Kurs bei annähernd 8.700 US-Dollar und nimmt Anlauf für das nächste Hoch (alle Werte in US-Dollar)

in die fremde Person. Bei der Übergabe ist nun eine große Gruppe anwesend, die die Transaktion beobachtet. Diese Gruppe bestätigt die Übergabe des Geldes von Person A zu Person B und kennt zusätzlich noch den korrekten Betrag, der überwiesen wurde. Die Identitäten der beteiligten Personen sind der Gruppe aber nicht bekannt.

Ein Resultat der Blockchain ist, dass man dadurch auf Mittelsmänner verzichten kann und damit Banken oder andere finanzielle Institutionen überflüssig gemacht werden. Kein Nutzer besitzt das Kassenbuch und jeder Eintrag ist irreversibel eingetragen, kann also nicht mehr getilgt werden. Aus diesem Grund ist die Blockchain auch eines der sichersten bisher bekannten Systeme überhaupt, denn eine Manipulation ist aufgrund des Designs nur sehr schwer möglich. Selbst wenn man mehrere Kopien der Datenbank besitzt, sie manipuliert und versucht, in das System einzustreuen, werden die Millionen Rechner die Legimität des veränderten Kassenbuches anzweifeln; erst mit mehr als 50 Prozent der Miner kann eine Fälschung gelingen („51-Prozent-Attacke“).

## Wo kommt das Geld her?

Viele fragen sich jetzt sicherlich: Wo kommen eigentlich die ganzen Coins her und wo ist der Gegenwert zu diesen digitalen Münzen?

Kryptowährungen werden einfach erzeugt, wobei „einfach“ Interpretationssache ist. Denn je mehr Leute ihre Rechenleistung für das Lösen der kryptografischen Hash-Funktion einsetzen, desto schwerer wird es gemacht, einen Bitcoin zu erzeugen. Diese Funktion ist aber zwingend notwendig, um einen neuen Block mit Transaktionsdaten in das Kassenbuch der Blockchain sicher eintragen zu können.

Als Belohnung für eine korrekt gelöste Hash-Funktion bekommen die Teilnehmer Bitcoins ausgezahlt, die „erstellt“ wurden. Man spricht hier vom sogenannten Mining oder Schürfen. Durch das künstliche Anheben des Schweregrades soll sichergestellt werden, dass keine einzelne Person willkürliche Änderungen vornehmen kann.

Der Aufwand ist enorm: Zeitweise vebrauchte das Bitcoin-Schürfen ganze 0,13 Prozent des weltweit verfügbaren Stroms und lag damit global gesehen vor 159 Ländern. Dieses energiehungrige Verfahren nennt man auch „Proof of Work“ (PoW) und es sichert über die kryptografischen Funktion die Transaktionsdaten in der Blockchain ab. Der Realwert eines Bitcoins entsteht erst aus der Bereitsschaft der Nutzer, für das Wertetransfersystem einer Blockchain von einem Staat emittiertes Geld, dafür auszugeben.

## Ökologische Alternativen?

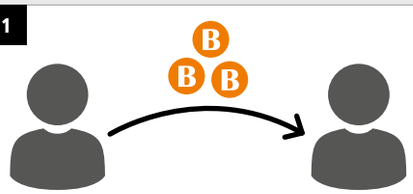
Neben dem Proof-of-Work-Mechanimus, der der Kryptowährung Bitcoin zugrunde liegt, gibt es noch weitere Verifizierungsmethoden innerhalb der Blockchain.

Beim Proof-of-Stake-Mechanismus wird die Stimmgewalt eines Miners von dessen Rechenleistung entkoppelt und stattdessen die im Besitz befindlichen Coins herangezogen. Dadurch entfällt der energiefressende Prozess, per mathematischer und kryptografischer Hash-Funktionen einen Block zu generieren. Der Proof of Stake funktioniert also so: Je größer der Coin-Anteil am gesamten Kryptogeldnetzwerk ist, desto mehr Datensätze kann dieser Miner verifizieren. Die geschürften Münzen, die dann bei der Verifizierung eines Blocks ausgegeben werden, sowie die Transaktionsgebühren, die eine (Distributed)-Denial-of-Ser-

# Wie eine Blockchain funktioniert

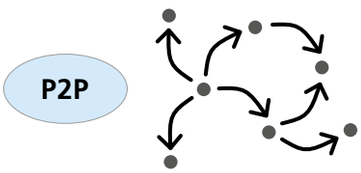
**Dezentral, ohne teure Mittelsmänner, demokratisch, fair und auch noch sicher? Hört sich einfach zu gut an. Wie kann das bloß funktionieren?**

**1**



Eine Transaktion wird durchgeführt, indem ein Wert von Person A auf Person B übertragen wird, die aber noch vom Netzwerk bestätigt werden muss.

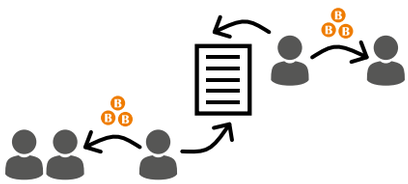
**2**



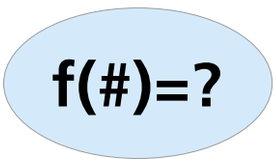

Dann wird diese Transaktion im Peer-to-Peer-Netzwerk der Blockchain gepostet, um dann von anderen Minern bestätigt zu werden, um Gültigkeit zu erlangen.

**3**

Alle zehn Minuten wird ein neuer Block erstellt, der alle Transaktionen und Informationen enthält – auch unsere.

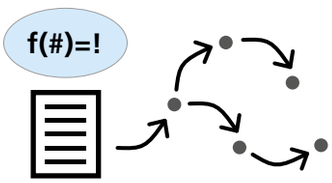


**4**




Nun wird eine kryptographische Hashfunktion gelöst, die den Regeln der Währung entsprechen. Hier konkurrieren die Miner untereinander, denn wer als erstes die Funktion löst, wird mit Coins belohnt.

**5**

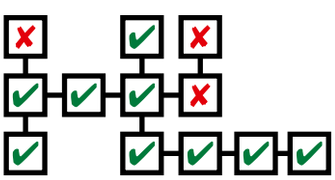



Sobald die Funktion gelöst wurde, muss sie von anderen Minern bestätigt werden, da bisher nur der Ersteller sie verifiziert hat. Dazu wird die Information über den gelösten Block wieder im P2P-Netzwerk gepostet.

**6**

Nur wenn der Block regelkonform gelöst wurde und bestimmte Daten enthält, wird er im Netzwerk als valide akzeptiert.

**7**



Ein valider Block wird schneller von anderen Minern verifiziert als ein manipulierter Block. Aus diesem Grund wächst die Blockchain mit korrekten Blöcken schneller als mit manipulierten. So ist die Blockchain sicher.

## Kurzes Bitcoin-Glossar

**Rund um den Bitcoin werden Interessierte mit viel Fachchinesisch konfrontiert. Die wichtigsten Begriffe im Überblick:**

### Altcoin
**Von „Alternative Coin“, bezeichnet alle anderen Kryptowährungen neben Bitcoin, wie beispielsweise Ether, Litecoin oder Monero.**

### ASIC und ASIC Miner
**Das Kürzel steht für „Application Specific Integrated Circuit“ und beschreibt einen Chip, der nur für eine bestimmte Aufgabe entworfen und produziert wurde. Ein ASIC Miner für Bitcoins enthält zum Beispiel Chips, die für das Lösen der SHA-256 Hash-Funktion optimiert wurden.**

### Dezentral autonom handelnde Organisation (DAO)
**Eine DAO ist ein Unternehmen, das auf der Blockchain und Smart Contracts basiert und keine menschliche Interaktion mehr benötigt, um geschäftsfähig zu sein. Kontrolliert wird eine DAO durch Algorithmen.**

### Fork
**Englisch für „Gabel“; bezeichnet die Abspaltung der Blockchain einer Kryptowährung in einem seperaten Entwicklungszweig. Kommt es zu einem Fork, sind nachkommende Datensätze mit den alten nicht mehr kompatibel. Bei Etherum kam es bereits zu einem Fork, als Coins im Wert von 50 Millionen Dollar gestohlen wurden.**

### Initial Coin Offering (ICO)
**Hierbei handelt es sich um eine unregulierte Crowdfundingaktion für Unternehmen, deren Geschäftsmodell auf der Blockchain basiert. Coins fungieren hier wie Aktien, die beim „Börsengang“ (Offering) verkauft werden und so Kapital erzeugen.**

### Ledger
**Engl. für Kassenbuch; beschreibt im Endeffekt nichts anderes als die unwiderruflich eingetragenen Datensätze (Transaktionen) in der Blockchain.**

### Private Key (privater Schlüssel)
**Nur mit dem privaten Schlüssel kann ein Nutzer auf ein Wallet (Bankkonto in der Blockchain) zugreifen und Transaktionen in der Blockchain durchführen. Vorstellen kann man sich das wie eine geheime PIN.**

### Satoshi
**Ein Satoshi ist die kleinste Untereinheit eines Bitcoin (ein millionstel Teil). Satoshi Nakamoto hatte 2008 in einem White Paper das Prinzip einer Kryptowährung vorgestellt und gilt deshalb als Erfinder des Bitcoin. Welche Person oder Personengruppe sich hinter dem Pseudonym verbirgt, ist unbekannt.**

### SHA-256
**Hierbei handelt es sich um die verwendete kryptografische Hash-Funktion in der Bitcoin-Blockchain. Das Kürzel steht für „Secure Hash Algorithm 256“; die Zahl repräsentiert die Länge des Hash-Wertes in Bit.**

### Smart Contract
**Smart Contracts sind Computerprotokolle, die Verträge darstellen oder überprüfen und auch die Vertragsverhandlungen oder -abwicklung technisch unterstützen. In der digitalen Rechteverwaltung (DRM) könnten diese Veträge die Lizenzen abbilden und durchsetzen. Bei Etherum werden mit solchen Verträgen ganze Unternehmen konstruiert, wie etwa die DAO (siehe weiter links).**

vice-Attacke (DDoS) verhindern sollen, werden dann zufällig und periodisch über die Coin-Besitzer ausgeschüttet. Dabei ist entscheidend, wie viel ein Miner besitzt und wie lange er über diesen Besitz verfügen konnte.

Alternativ gibt es noch mehr Mechanismen, die ähnlich zum Proof of Stake die Energiekosten der Miner auf ein Minimum reduzieren sollen. Zu nennen wären hier neben dem Proof of Burn (PoB) noch der Proof of Capacity (PoC; auch Proof of Space). Die Verifizierung weiterer Blöcke innerhalb der Blockchain wird beim PoB erreicht, indem Coins der Kryptowährung zerstört werden und dadurch dem Miner ein Wert gegeben wird, der ähnlich dem Hash-Wert beim PoW ist. Beim PoC wird dagegen auf Speicherkapazität gesetzt anstatt auf Rechenleistung wie beim Proof of Work.

## Die Zukunft des Geldes?

Was kann man nun mit Kryptowährungen und der Blockchain alles anfangen? Anhand der Blockchain lässt sich wie erwähnt genau klären, wie die Besitzverhältnisse digitaler Produkte sind. Ein Beispiel: Bis jetzt kann man eine MP3 beliebig oft kopieren, ohne dass man einen Unterschied zum Original ausmachen kann. Dank der Blockchain ist nun direkt erkennbar, welche MP3 das Original ist und welche nur eine Kopie, da in den Blöcken die Information zugrunde liegt. Man spricht hier vom Internet der Werte (Internet of Value), da jedem digitalen Produkt nun ein Wert zugeordnet werden kann.

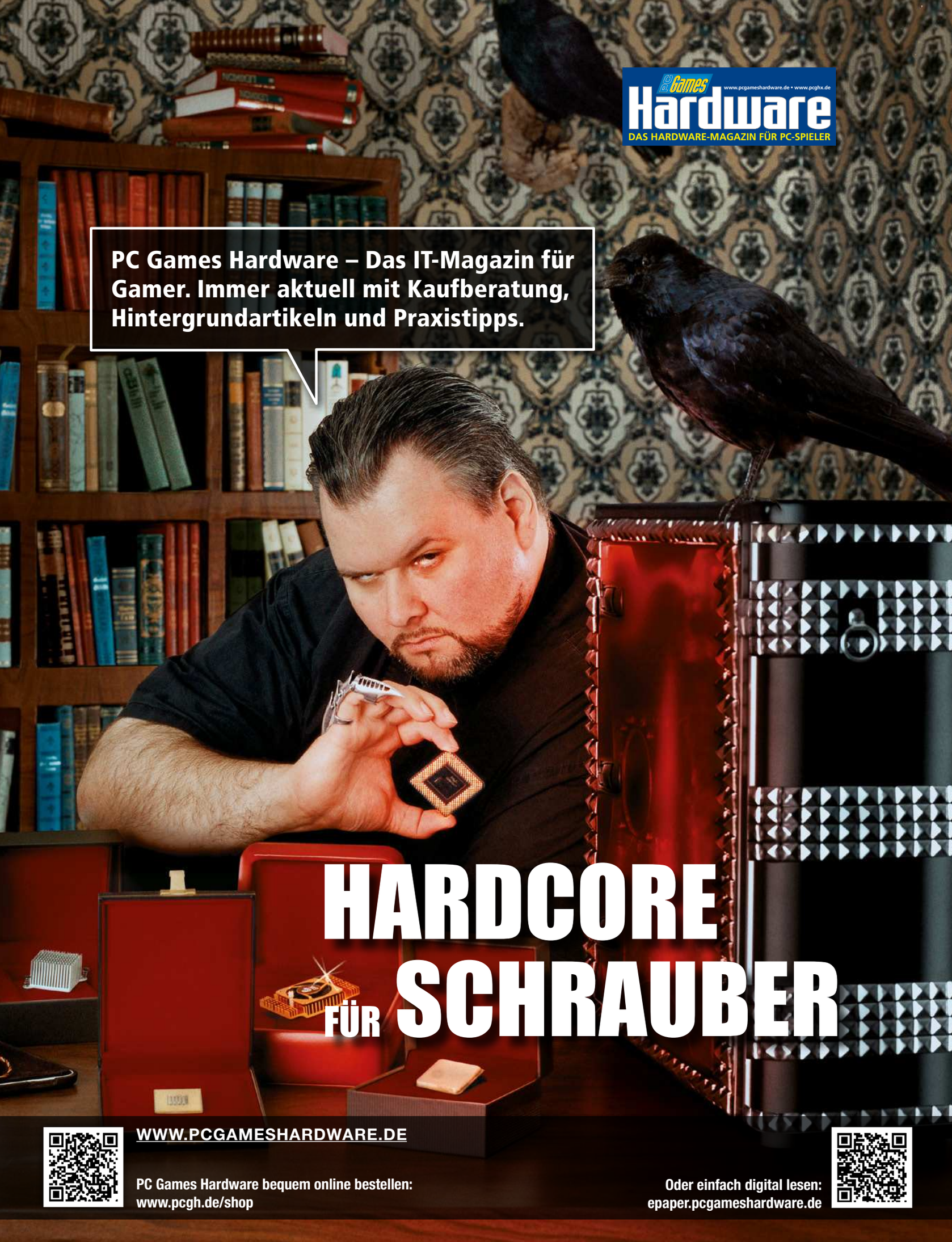
Die Möglichkeiten der Blockchain sind also vielfältig, angefangen beim sicheren, schnellen und günstigem Transfer von Finanzen durch Kryptowährungen wie Bitcoin oder Etherum. Durch das manipulationssichere Design der Blockchain kann diese Technologie aber auch in einer Wahl Anwendung finden. Unregelmäßigkeiten bei der Stimmauszahl würden damit der Vergangenheit angehören, da einerseits viele Computer über eine hinterlegte Kopie der Blockchain (und damit der abgegebenen Stimmen) verfügen und so jeder Gang zur Wahlurne zunächst verifiziert werden muss. Zudem kann jeder öffentlich einsehen, wie und für wen gewählt wurde, ohne dass dabei die Identität der Wähler zur Diskussion stehen würde.

## Neue Möglichkeiten

Natürlich können auch weitere Wertgegenstände mit der Blockchain kombiniert werden, beispielsweise mit Musik. Statt einem Service einen monatlichen Beitrag zu zahlen, der auf die Künstler aufgeteilt wird, könnte man Content-Schaffende so viel fairer und direkter entlohnen.

Die Blockchain könnte auch unser Energienetz umkrempeln. Im Moment wird Strom noch zentral erzeugt und verteilt und man merkt bereits, dass aufgrund erneuerbarer Energien eine dezentrale Gewinnung und Verteilung unumgänglich ist. Mittels Blockchain könnten private Haushalte den erzeugten Strom direkt und automatisch in der Nachbarschaft verkaufen oder bei eigenem Bedarf im näheren Umfeld einkaufen. Alles wäre manipulationssicher in den Datensätzen hinterlegt. In Australien und New York laufen bereits erste Versuche.

Die Technologie der Blockchain ist innovativ und kann auch in anderen Aspekten unseres Lebens Einzug finden. Dann würden wir nicht nur unabhängiger von großen Institutionen und Firmen, sondern könnten Werte auch fairer und direkter entlohnen.


PC Games Hardware
www.pcgameshardware.de • www.pcghx.de
DAS HARDWARE-MAGAZIN FÜR PC-SPIELER
PC Games Hardware – Das IT-Magazin für Gamer. Immer aktuell mit Kaufberatung, Hintergrundartikeln und Praxistipps.
HARDCORE FÜR SCHRAUBER
WWW.PCGAMESHARDWARE.DE
PC Games Hardware bequem online bestellen: www.pcgh.de/shop
Oder einfach digital lesen: epaper.pcgameshardware.de

# Kompromisslos Kompakt

## DARK ROCK SLIM

Der Dark Rock Slim verfügt über eine besonders hohe Kühlleistung von 180W TDP und arbeitet dabei dank des Silent Wings 3 120mm PWM Lüfters nahezu unhörbar leise. Gleichzeitig ist er sehr kompakt gebaut, und bietet dadurch jede Menge Platz für außergewöhnliche RAM-Riegel, ohne Speicherbänke zu blockieren. All dies ergibt in Kombination mit der legendären be quiet!-Zuverlässigkeit die perfekte Wahl für High-End-Systeme mit begrenztem Platzangebot.

- Platzsparendes Design
- Speicherbänke werden nicht blockiert
- Besonders hohe Kühlleistung von 180W TDP
- Nahezu unhörbarer Silent Wings 3 120mm PWM Lüfter
- Spezielle Beschichtung mit Keramik-Partikeln sichert perfekte Wärmeübertragung

Für mehr Informationen besuchen Sie **bequiet.com**

GERMANY'S NO. 1* PSU MANUFACTURER

*GfK 2007-2019

**Erhältlich bei:** alternate.de · arlt.de · bora-computer.de · caseking.de · computeruniverse.net conrad.de · hiq24.de · e-tec.at · galaxus.ch · mindfactory.de · notebooksbilliger.de · reichelt.de

be quiet!®